



# **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

## **FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

### **ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES**

#### **IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA SUPERVISIÓN DE NIÑOS ENTRE 2 A 4 AÑOS USANDO VISIÓN ARTIFICIAL.**

#### **TRABAJO DE TITULACIÓN**

**Tipo:** DISPOSITIVO TECNOLÓGICO

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES**

**AUTOR:** JUAN ALEXANDER LUDEÑA CHICA

**TUTOR:** ING. PABLO EDUARDO LOZADA YÁNEZ

Riobamba – Ecuador

**2019**

**©2019, Juan Alexander Ludeña Chica**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

# ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

## FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

### ESCUELA DE ING. ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

El tribunal de Trabajo de Titulación certifica que: El trabajo de investigación: IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD PARA SUPERVISIÓN DE NIÑOS ENTRE 2 A 4 AÑOS USANDO VISIÓN ARTIFICIAL, de responsabilidad del señor Juan Alexander Ludeña Chica, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

Ing. Washington Gilberto Luna E.

**Decano de la Facultad de  
Informática y Electrónica**

\_\_\_\_\_

Ing. Freddy Enrique Chávez V.

**Director de escuela de  
Ingeniería Electrónica en  
control y Redes Industriales**

\_\_\_\_\_

Ing. Pablo Eduardo Lozada Y.

**Director del Trabajo de  
Titulación**

\_\_\_\_\_

Ing. José Luis Tinajero L.

**Miembro del Trabajo de  
Titulación**

\_\_\_\_\_

Yo, Juan Alexander Ludeña Chica soy responsable de todos los resultados, ideas y modos expuestos en este Trabajo de Titulación y del patrimonio intelectual del trabajo de titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

---

Juan Alexander Ludeña Chica

**ESTUDIANTE**

## **DEDICATORIA**

Quiero dedicar este trabajo de investigación a mi familia, mis padres los cuales con su cariño, confianza y años de trabajo dieron todo de ellos para apoyarme en todo lo que me proponía. Gracias a ellos y su apoyo incondicional pude lograr este objetivo, en el futuro quisiera ser como ellos humildes, sencillos y trabajadores.

Mi hermano quien estuvo más pendiente de mí día a día, me dio ánimo y consejos en momentos difíciles de mi vida cuando pensé decaer él estuvo ahí para ayudarme y no darme por vencido en esta lucha que hoy rindieron sus frutos.

## **AGRADECIMIENTO**

El agradecimiento infinito a mis padres y mi hermano que sin ellos no habría emprendido esta carrera universitaria y nunca dejaron de creer en mí. A mis amigos José y Stalin que son quienes más he compartido en estos años de estudio han llegado a ser como mi segunda familia.

Gracias a los maestros de la carrera de Ing. Electrónica quienes impartieron sus conocimientos a lo largo de mis estudios. En especial a mi tutor y miembro de tribunal del proyecto de titulación los ingenieros Pablo y José Luis quienes con su ayuda y recomendaciones contribuyeron a la finalización de la mencionada investigación.

## ÍNDICE GENERAL

ÍNDICE DE ANEXOS .....	ix
ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE GRÁFICOS.....	xiii
ABREVIATURAS.....	xiv
RESUMEN.....	xvi
ABSTRACT .....	xvii
INTRODUCCIÓN .....	1

## CAPÍTULO I

1. MARCO TEÓRICO .....	4
1.1 Sistemas de seguridad para niños .....	4
1.2 Visión Artificial .....	5
1.2.1 Aplicaciones y métodos de visión artificial.....	6
1.3 Sistemas de visión artificial .....	9
1.4 Fases del sistema de visión artificial .....	11
1.5 Componentes del sistema de procesamiento .....	12
1.6 Hardware utilizado en visión artificial.....	15
1.6.1 Cámaras .....	15
1.6.2 Tipos de cámaras .....	15
1.6.3 Raspberry Pi .....	19
1.6.4 Arduino .....	20
1.7 Software de programación .....	21
1.7.1 Python .....	21
1.7.1.1 Características .....	22
1.7.2 OpenCv .....	22
1.7.3 SIFT & FLANN .....	23

1.7.4	<i>HARR Cascade</i> .....	25
-------	---------------------------	----

## CAPITULO II

2.	<b>MARCO METODOLÓGICO</b> .....	26
2.1	<b>Selección de Dispositivos</b> .....	26
2.2	<b>Hardware</b> .....	26
2.2.1	<i>Raspberry Pi 3</i> .....	26
2.2.2	<i>Arduino Uno</i> .....	27
2.2.3	<i>Módulo GSM SIM 900</i> .....	28
2.2.4	<i>Cámara Logitech C270</i> .....	29
2.2.5	<i>Convertidor DC-DC Step Down</i> .....	30
2.2.6	<i>Módulo Relé de un canal</i> .....	31
2.2.7	<i>Módulo DF Player Mini</i> .....	33
2.2.8	<i>Parlantes</i> .....	34
2.2.9	<i>Baliza</i> .....	34
2.2.10	<i>Pantalla</i> .....	35
2.3	<b>Software</b> .....	35
2.3.1	<i>Open CV Library</i> .....	35
2.3.2	<i>Qt Designer</i> .....	36
2.4	<b>Descripción del proceso</b> .....	36
2.5	<b>Desarrollo de la interfaz gráfica</b> .....	37
2.5.1	<i>Entrenamiento de Haar Cascade</i> .....	40
2.5.2	<i>Comprobación del entrenamiento</i> .....	44
2.6	<b>Programación Python</b> .....	45
2.7	<b>Sistema detección y alarma</b> .....	46
2.8	<b>Programación en Arduino</b> .....	47
2.7	<b>Diagrama de control</b> .....	48



## CAPÍTULO III

<b>3.</b>	<b>Marco de Resultados.....</b>	<b>50</b>
<b>3.1</b>	<b>Pruebas del clasificador .....</b>	<b>50</b>
<b>3.1.1</b>	<b><i>Mediante Raspberry Pi en espacios controlados .....</i></b>	<b>50</b>
<b>3.1.2</b>	<b><i>Mediante computadora en espacios controlados .....</i></b>	<b>52</b>
<b>3.1.3</b>	<b><i>Mediante Raspberry Pi en espacios no controlados .....</i></b>	<b>54</b>
<b>3.1.4</b>	<b><i>Mediante computadora en espacios no controlados .....</i></b>	<b>57</b>
<b>3.2</b>	<b>Comparación de resultados entre Raspberry y computador .....</b>	<b>59</b>
<b>3.2.1</b>	<b><i>Espacios controlados .....</i></b>	<b>59</b>
<b>3.2.2</b>	<b><i>Espacios no controlados.....</i></b>	<b>60</b>
<b>3.3</b>	<b>Prueba de detección de objetos a distancia .....</b>	<b>60</b>
<b>3.3.1</b>	<b><i>Detección de objetos en espacios controlados .....</i></b>	<b>60</b>
<b>3.3.2</b>	<b><i>Detección de objetos en espacios no controlados .....</i></b>	<b>61</b>
<b>3.4</b>	<b>Censo de carga.....</b>	<b>61</b>
<b>3.5</b>	<b>Presupuesto.....</b>	<b>63</b>
	<b>CONCLUSIONES.....</b>	<b>64</b>
	<b>RECOMENDACIONES.....</b>	<b>65</b>
	<b>GLOSARIO</b>	
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## **ÍNDICE DE ANEXOS**

**Anexo A** Creación del prototipo en Solid Works

**Anexo B** Pruebas de diseño de la interfaz gráfica

**Anexo C** Prueba de algoritmo

**Anexo D** Toma de muestras

**Anexo E** Prueba final del sistema de supervisión

**Anexo F** Prueba de detección de objetos en espacios controlados

**Anexo G** Prueba de detección de distancia en espacios no controlados

**Anexo H** Esquemático de control

**Anexo I** Esquemático de Board

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b> Fases del sistema de visión artificial .....	11
<b>Tabla 2-1:</b> Propiedades de las imágenes digitales .....	14
<b>Tabla 3-1:</b> Características de los modelos de Raspberry Pi .....	19
<b>Tabla 4-1:</b> Características de los Arduinos .....	20
<b>Tabla 5-1:</b> Componentes Principales de OpenCV .....	22
<b>Tabla 1-2:</b> Conexión de pines del Arduino UNO .....	27
<b>Tabla 2-2:</b> Características de módulo GSM SIM900 .....	29
<b>Tabla 3-2:</b> Especificaciones técnicas de la cámara Logitech C270 .....	30
<b>Tabla 4-2:</b> Características del Convertidor DC-DC Setp Down Kim 0551 .....	31
<b>Tabla 5-2:</b> Especificaciones técnicas del módulo relé de un canal .....	32
<b>Tabla 6-2:</b> Características del módulo DF Player Mini .....	33
<b>Tabla 7-2:</b> Descripción de botones empleados en la interfaz gráfica .....	37
<b>Tabla 8-2:</b> Comandos empleados en la programación del Arduino UNO .....	48
<b>Tabla 1-3:</b> Prueba del sistema mediante Raspberry Pi en espacios controlados .....	50
<b>Tabla 2-3:</b> Prueba de funcionamiento mediante computadora en espacios controlados .....	52
<b>Tabla 3-3:</b> Prueba de funcionamiento mediante Raspberry Pi en espacios no controlados .....	55
<b>Tabla 4-3:</b> Prueba del sistema mediante computadora en espacios no controlados .....	57
<b>Tabla 5-3:</b> Resultado de pruebas de funcionamiento en espacios controlados .....	59
<b>Tabla 6-3:</b> Resultado de pruebas de funcionamiento en espacios no controlados .....	60
<b>Tabla 7-3:</b> Detección de objetos en espacios controlados .....	60
<b>Tabla 8-3:</b> Detección de objetos en espacios no controlados .....	61
<b>Tabla 9-3:</b> Censo de potencia .....	62
<b>Tabla 10-3:</b> Tiempo de duración de baterías estándar .....	62
<b>Tabla 11-3:</b> Costos .....	63

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b> Sistema de seguridad para niños .....	4
<b>Figura 2-1:</b> Diagrama de procesamiento de imagen .....	5
<b>Figura 3-1:</b> Categorías principales de visión artificial .....	6
<b>Figura 4-1:</b> Imágenes usadas en el guiado .....	7
<b>Figura 5-1:</b> Imágenes guiadas por patrones geométricos .....	7
<b>Figura 6-1:</b> Identificación mediante visión artificial.....	7
<b>Figura 7-1:</b> Sistema de medición con visión artificial .....	8
<b>Figura 8-1:</b> Sistema de inspección por visión artificial.....	8
<b>Figura 9-1:</b> Esquema de imagen binaria .....	9
<b>Figura 10-1:</b> Sistema de visión 1D.....	9
<b>Figura 11-1:</b> Sistema de visión 2D.....	10
<b>Figura 12-1:</b> Sistema de visión lineal.....	10
<b>Figura 13-1:</b> Sistema de visión 3D.....	10
<b>Figura 14-1:</b> Fases del sistema de visión artificial .....	11
<b>Figura 15-1:</b> Procesamiento de imagen.....	12
<b>Figura 16-1:</b> Tipos de iluminación.....	12
<b>Figura 17-1:</b> Sensor CCD.....	13
<b>Figura 18-1:</b> Procesamiento de imágenes .....	14
<b>Figura 19-1:</b> Representación de imágenes .....	15
<b>Figura 20-1:</b> Cámara lineal .....	16
<b>Figura 21-1:</b> Píxeles en cámaras lineales .....	16
<b>Figura 22-1:</b> Cámara matricial .....	16
<b>Figura 23-1:</b> Proceso cámara de color.....	17
<b>Figura 24-1:</b> Sensor CCD.....	17
<b>Figura 25-1:</b> Sensor CMOS .....	18
<b>Figura 26-1:</b> Raspberry Pi.....	19
<b>Figura 27-1:</b> Ejemplo de lenguaje Python.....	21
<b>Figura 32-1:</b> Características de Python .....	22
<b>Figura 33-1:</b> Algoritmo SIFT & FLANN para detección de objetos .....	23
<b>Figura 34-1:</b> Clasificación de objetos mediante SIFT .....	24
<b>Figura 36-1:</b> Clasificador de patrones mediante HAAR Cascade.....	25
<b>Figura 1-2:</b> Placa Raspberry Pi 3 Modelo B .....	26
<b>Figura 2-2:</b> Arduino Uno .....	27
<b>Figura 3-2:</b> Módulo GSM SIM900 .....	28

<b>Figura 4-2:</b> Cámara Logitech C270 .....	29
<b>Figura 5-2:</b> Convertidor DC-DC Step Down .....	30
<b>Figura 6-2:</b> Módulo Relé de un canal.....	31
<b>Figura 7-2:</b> Módulo DF Player Mini.....	33
<b>Figura 8-2:</b> Parlante USB.....	34
<b>Figura 9-2:</b> Baliza .....	34
<b>Figura 10-2:</b> Pantalla.....	35
<b>Figura 11-2:</b> Identificador OpenCV .....	35
<b>Figura 12-2:</b> Interfaz Qt Designer.....	36
<b>Figura 13-2:</b> Proceso de supervisión.....	36
<b>Figura 14-2:</b> Diagrama de la interfaz gráfica .....	37
<b>Figura 15-2:</b> Código HTML de los botones de la interfaz gráfica .....	38
<b>Figura 16-2:</b> Código HTML de los labels de la interfaz gráfica .....	39
<b>Figura 17-2:</b> Interfaz gráfica terminada .....	40
<b>Figura 18-2:</b> Pantalla principal del Cascade Trainer GUI.....	40
<b>Figura 19-2:</b> Recopilación de imágenes y videos .....	41
<b>Figura 20-2:</b> Selección de imágenes positivas .....	41
<b>Figura 21-2:</b> Imágenes positivas guardadas .....	42
<b>Figura 22-2:</b> Configuración del entrenador.....	43
<b>Figura 23-2:</b> Final del entrenamiento.....	43
<b>Figura 24-2:</b> Archivo XML generado al final del Cascade .....	44
<b>Figura 25-1:</b> Comprobación del entrenamiento .....	44
<b>Figura 26-2:</b> Diagrama de programación en Python .....	45
<b>Figura 27-2:</b> Algoritmo del sistema de detección y alarma .....	46
<b>Figura 28-2:</b> Diagrama de funcionamiento del algoritmo .....	47
<b>Figura 29-2:</b> Esquema de diagrama de control .....	48

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-3:</b> Tiempo de respuesta mediante Raspberry Pi en espacios controlados .....	50
<b>Gráfico 2-3:</b> Número de objetos mediante Raspberry Pi en espacios controlados.....	51
<b>Gráfico 3-3:</b> Distancia de objetos mediante Raspberry Pi en espacios controlados.....	51
<b>Gráfico 4-3:</b> Tiempo de respuesta de llamada en Raspberry Pi en espacios controlados .....	52
<b>Gráfico 5-3:</b> Tiempo de respuesta por computador en espacios controlados .....	53
<b>Gráfico 6-3:</b> Número de objetos por computador en espacios controlados.....	53
<b>Gráfico 7-3:</b> Distancia de objetos mediante computadora en espacios controlados .....	54
<b>Gráfico 8-3:</b> Tiempo de respuesta de llamada por computador en espacios controlados .....	54
<b>Gráfico 9-3:</b> Tiempo de respuesta mediante Raspberry Pi en espacios no controlados.....	55
<b>Gráfico 10-3:</b> Número de objetos por Raspberry Pi en espacios no controlados.....	55
<b>Gráfico 11-3:</b> Distancia de objetos mediante Raspberry Pi en espacios no controlados.....	56
<b>Gráfico 12-2:</b> Tiempo de respuesta por Raspberry Pi en espacios no controlados .....	56
<b>Gráfico 13-3:</b> Tiempo de respuesta computadora en espacios no controlados .....	57
<b>Gráfico 14-3:</b> Número de objetos por computadora en espacios no controlados.....	58
<b>Gráfico 15-3:</b> Distancia de objetos mediante computadora en espacios no controlados.....	58
<b>Gráfico 16-3:</b> Tiempo de respuesta mediante computadora en espacios no controlados.....	59

## ABREVIATURAS

<b>1D:</b>	Una dimensión
<b>2D:</b>	Dos dimensiones
<b>3D:</b>	Tres dimensiones
<b>CCD:</b>	Charge Coupled Device / Dispositivo acoplado de carga
<b>CMOS:</b>	Complementary Metal Oxide Semiconductor / Semiconductor de óxido de metal complementario
<b>CPU:</b>	Central Processing Unit / Unidad de Procesamiento Central
<b>FLANN:</b>	Fast Library for Approximate Nearest Neighbors / Biblioteca rápida para los vecinos más cercanos aproximados
<b>GIF:</b>	Graphics Interchange Format / Formato de Intercambio de Gráficos
<b>GPIO:</b>	General Purpose Input-Output / Entrada-salida de uso general
<b>GPU:</b>	Graphics Processing Unit / Unidad De Procesamiento Gráfico
<b>GUI:</b>	Graphical User Interface / Interfaz Gráfica De Usuario
<b>HDMI:</b>	High-Definition Multimedia Interface / Interfaz Multimedia De Alta Definición
<b>I<sup>2</sup>C:</b>	Inter-Integrated Circuit / Circuito Inter-Integrado
<b>ISO:</b>	International Organization for Standardization / Organización Internacional para la Estandarización.
<b>ITL:</b>	Transferencia Inter línea
<b>LED:</b>	Light-Emitting Diode / Diodo Emisor De Luz
<b>MHz:</b>	Unidad de Medida de Frecuencia
<b>OpenCv:</b>	Open Source Computer Vision / Visión de computadora de código abierto

<b>PWM:</b>	Pulse-Width Modulation / Modulación por Ancho de Pulsos
<b>RAM:</b>	Random Access Memory / Memoria de Acceso Aleatorio
<b>RCA:</b>	Radio Corporation of America / Radio Corporación de America
<b>RGB:</b>	Red, Green, Blue / Rojo, Verde y Azul
<b>SIFT:</b>	Scale-Invariant Feature Transform / Transformación de Característica Invariante De Escala
<b>SoC:</b>	System On Chip / Sistema en un Chip
<b>SPI:</b>	Serial Peripheral Interface / Interfaz Periférica Serial
<b>SURF:</b>	Speeded-Up Robust Features / Características Robustas Aceleradas
<b>UART:</b>	Universal Asynchronous Receiver-Transmitter / Receptor-Transmisor Asíncrono Universal
<b>USB:</b>	Universal Serial Bus / Bus Universal en Serie
<b>μm:</b>	Micrómetro, Micrón o Micra / Unidad de longitud equivalente a una milésima parte de un milímetro



## RESUMEN

Debido a la gran cantidad de accidentes que se ha registrado en los hogares, se implementó un sistema de seguridad para niños de 2 a 4 años usando visión artificial, utilizando elementos como Raspberry Pi 3, Arduino UNO, módulo SIM 900, DF Player, relé de un canal, pantalla, baliza, parlantes entre otros para la construcción del prototipo. El desarrollo del sistema de detección se basó en el algoritmo llamado “Haar Cascade”, la creación de la interfaz se realizó mediante la interfaz gráfica de usuario (GUI) Cascade Trainer. Partiendo de esto, se desarrolló el sistema que permite detectar objetos peligrosos para los niños. Cuando el sistema detecta un objeto peligroso como por ejemplo tijeras, cuchillos etc. activa una alarma de emergencia que inmediatamente emite una llamada al número registrado en la base de datos. El algoritmo de Cascade permitió crear una base de datos de imágenes positivas y negativas en este caso las imágenes positivas serán los objetos a detectar. La interfaz gráfica es visualizada por el usuario permitiendo manipular seis botones, cada uno cumple funciones específicas que son: iniciar cámara, pausa, detección, activar alarma, reinicio y probar GSM. Según las pruebas realizadas, se establece que el sistema funciona de manera óptima en espacios controlados, mientras que en espacios no controlados los parámetros varían en un 0,38% el tiempo de respuesta del sistema, en 2% el número de objetos reconocidos, en 1% la distancia de los objetos a detectar y un 2% el tiempo de respuesta de llamada, es decir el sistema funciona de manera óptima, pero tarda en enviar las respuestas.

**Palabras clave:** <SISTEMA DE SUPERVISIÓN> <SEGURIDAD PARA NIÑOS> <VISIÓN ARTIFICIAL> <DETECCIÓN DE OBJETOS> <ALGORITMO DE CLASIFICACIÓN> <ALGORITMO HAAR CASCADE> <DESARROLLO DE INTERFAZ GRÁFICA>

## **ABSTRACT**

Due to the large number of accidents that have been registered in homes, a security system was implemented for children from 2 to 4 years using artificial vision, using elements such as Raspberry Pi 3. Arduino UNO. SIM module 900. DF Player. relay of a channel, screen, beacon, speakers among others for the construction of the prototype. The development of the detection system was based on the algorithm called "Haar Cascade", the creation of the interface was made using the graphical user interface (GUI) Cascade Trainer. Based on this, the system was developed to detect dangerous objects for children. When the system detects a dangerous object such as scissors, knives, etc. activates an emergency alarm that immediately issues a call to the number registered in the database. The Cascade algorithm allowed to create a database of positive and negative images in this case the positive images will be the objects to be detected. The graphical interface is visualized by the user allowing to manipulate six buttons, each one fulfils specific functions that are: start camera, pause, detection, activate alarm, reboot and test GSM. According to the tests carried out, it is established that the system works optimally in controlled spaces, while in uncontrolled spaces the parameters vary by 0.38% the response time of the system, in 2% the number of objects recognized, in 1% the distance of the objects to be detected and 2% the call response time, that is, the system works optimally, but it takes time to send the answers.

**KEYWORDS:** <SUPERVISION SYSTEM <CHILD SAFETY> <ARTIFICIAL VISION>  
<DETECTION OF OBJECTS> <CLASSIFICATION ALGORITHM> <HAAR CASCADE  
ALGORITHM> <GRAPHICAL INTERFACE DEVELOPMENT>

## INTRODUCCIÓN

Uno de los accidentes comunes en menores de edad es el que se produce por la aspiración de un objeto pequeño y su alojamiento en las vías aéreas, lo cual causa una alta mortalidad y morbilidad en el mundo entero, esto significa que muchos niños especialmente en edades pediátricas entre los 2 y 3 años, mueren asfixiados y otros pueden quedar con problemas neurológicos o pulmonares posteriores a este evento.

Los niños entre 0 a 4 años son las víctimas prevalentes por numerosas razones: curiosidad, exploración del ambiente por medio de la boca, deglución inmadura, propensión de los niños a la distracción y a jugar mientras comen, y falta de capacidad cognitiva para distinguir objetos comestibles de los que no lo son.

Iván Stand Niño, presidente de la Asociación colombiana de Neumología Pediátrica, asegura que la aspiración de cuerpos extraños “es responsable del 25% de las muertes accidentales en niños menores de 4 años. De hecho, más del 90 por ciento de los casos ocurren en esta edad”. (Chilena y Pediatría 2010). Según el diario “EL COMERCIO” en Ecuador en el año 2013 se registró 300 fallecidos por estos incidentes. De ellos, el 39% fue por asfixia, es decir, 116 muertes.

En la actualidad existen dispositivos comerciales como cámaras móviles de video vigilancia panorámica y receptora de imagen y sonido, para ver y oír al bebé en cualquier lugar y en la oscuridad a continuación se detallan algunos:

Baby Monitor 3G: El cual es una APP para IOS, disponible desde el 2013, en él se configura el servicio de monitoreo con dos dispositivos, por ejemplo, un iPhone, iPod, iPad.

Baby Ping: Compuesto por una cámara de video 640 x 480 pixeles, con micrófono incorporado, colocada en la habitación del bebe, permite monitorear en tiempo real las actividades del niño(a), estos equipos permiten la conexión vía Wifi a un dispositivo iPhone, iPod o iPad touch.

A nivel nacional en la Universidad de Guayaquil se tiene un sistema de control y monitoreo de bebés basados en Open Source, conformado por un conjunto de sensores, una tarjeta Arduino, una Raspberry Pi y el software desarrollado en lenguajes de código abierto, posibilitan la creación de un dispositivo de bajo costo, que se pueda adaptar a las funcionalidades mencionadas.

Los estudios encontrados no cuentan con un sistema de supervisión automático, donde el infante esté en constante vigilancia por medio de un sistema de visión artificial donde identificará objetos que son o no perjudiciales para su salud y así evitar en lo máximo este tipo de accidentes debido a que el deseo de un niño por descubrir el mundo puede hacer que, en ocasiones, lleve a su boca objetos que ponen en peligro su vida y es por esto que se desea implementar un sistema de seguridad para supervisión de niños de 2 a 4 años usando visión artificial.

## **PLANTEAMIENTO DEL PROBLEMA**

¿Cómo implementar un sistema de seguridad para supervisión de niños entre 2 a 4 años usando Visión Artificial?

¿Cómo funciona un sistema de supervisión para niños en temprana edad?

¿Cuáles son los requerimientos que debe cumplir el sistema a implementar?

¿Con que elementos de hardware y software se llevará a cabo la implementación del proyecto?

¿El sistema tiene un buen rendimiento?

## **JUSTIFICACIÓN**

La tecnología basada en visión artificial pretende capturar la información visual del entorno físico para extraer características relevantes visuales, utilizando procedimientos automáticos. Según Marr, “Visión es un proceso que produce a partir de imágenes del mundo exterior una descripción útil para el observador y no tiene información irrelevante”.

A medida que el niño va creciendo adquiere la capacidad de moverse, girarse, observar y llevarse todo a la boca. Cualquier objeto a su alcance, aumentará el riesgo de atragantamiento a partir de los cuatro meses de edad. Los trocitos de comida, las piezas de algunos juguetes que pueda alcanzar o los objetos le den otros niños de su entorno pueden ser causa de un atragantamiento.

Existe diversos objetos que pueden causar asfixia por ejemplo monedas, botones, juguetes con piezas pequeñas, pelotas o bolas pequeñas, canicas, globos, pequeñas baterías y los alimentos causan más del 50% de los episodios de asfixia y estos accidentes suelen ocurrir de forma rápida e inesperada, creando mucha angustia en los padres de ahí la importancia de saber actuar ante esta situación.

El diseño de un correcto sistema de iluminación es importante para la implementación de un sistema de visión artificial. Una adecuada iluminación permitirá resaltar los rasgos de interés del objeto, reducir la complejidad de la imagen a analizar y mejorar el tiempo de respuesta del procesamiento digital.

## **OBJETIVOS**

### **OBJETIVO GENERAL**

- Implementar un sistema de seguridad para supervisión de niños entre 2 a 4 años usando visión artificial.

### **OBJETIVOS ESPECÍFICOS**

- Analizar el funcionamiento de un sistema de supervisión para niños en temprana edad.
- Especificar los requerimientos que debe cumplir el sistema a implementar.
- Seleccionar los dispositivos de hardware y software que se llevará a cabo en la implementación del proyecto.
- Evaluar el rendimiento del sistema.

## CAPÍTULO I

### 1. MARCO TEÓRICO

#### 1.1 Sistemas de seguridad para niños



**Figura 1-1:** Sistema de seguridad para niños

**Fuente:** <http://www.areadeluxe.com/estores-y-cortinas-con-sistema-de-seguridad-para-ninos-child-safety/>

Los niños de 2 a 4 años de edad comienzan a desarrollar curiosidad por el entorno que los rodea, pero existen elementos habituales en el hogar que pueden convertirse en riesgos potenciales para la salud de los niños. (Simón. 2017)

Los sistemas de seguridad son vitales para la protección de los niños, en la actualidad existen sistemas que se utiliza en hogares, escuelas y vehículos. Son mecanismos enfocados principalmente a evitar robos, actos vandálicos y otros percances del exterior. (Simón. 2017)

**Sistemas de retención infantil:** Son dispositivos que se emplea en los vehículos conocidos comúnmente como silleas o asientos móviles, el sistema utiliza un cinturón de seguridad que mantiene seguro al menor en los viajes y evita que el niño sea arrojado afuera del vehículo en caso de colisión.

**Sistema de video vigilancia:** Son mayormente usados para vigilancia de empresas, negocios y hogares, pero son escasos para el monitoreo de niños. Los sistemas que se dedican a la vigilancia de niños se conocen como vigila bebés, pero están dirigidos solo a niños de 0 a 2 años se pueden nombrar algunos como, por ejemplo: Availand Follow Baby, Motorola MBP 36SC entre otros.

**Sistemas de seguridad para bebés:** Comúnmente conocidos como intercomunicadores, permite que los padres puedan estar en constante vigilancia de los niños y auxiliarlos en caso que sea

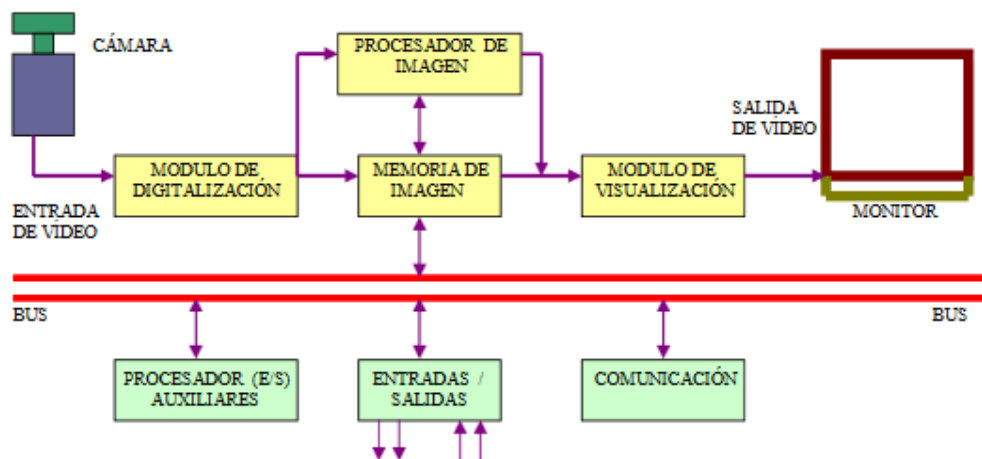
necesario ya que cuentan con mecanismos que permiten escuchar y ver en tiempo real. Existen varios tipos de dispositivos según la emisión de señal, los cuales se describen a continuación:

- **Analógicos:** Generan interferencia al estar cerca de otros dispositivos, suelen ser más económicos que los digitales.
- **Digitales:** Emplean tecnología inalámbrica que reducen las interferencias o ruidos externos, son más confiables.
- **De sonido:** Son bidireccionales por lo cual están diseñados específicamente para escuchar al bebe y hablar con él.
- **De video:** la mayoría de sistemas que emplean este tipo son inalámbricas y la cámara están diseñada para tener un alcance de 330m, por lo cual el niño esta monitoreado todo el tiempo.

## 1.2 Visión Artificial

Permite a las maquinas obtener información y transformarla a imágenes. Es aplicada en industrias para mantener un control de calidad, otra manera muy importante de emplearla es en robots permitiendo convertir la información en patrones de realidad aumentada. (García, E. 2016)

Es un campo que se deriva de la inteligencia artificial, mediante un conjunto de procesos y técnicas realiza el análisis, obtención y procesamiento de información la cual es transformada, memorizada e interpretada en imágenes digitales.



**Figura 2-1:** Diagrama de procesamiento de imagen

Fuente: <http://www.etitudela.com/celula/downloads/visionartificial.pdf>

### La visión artificial permite:

- Automatizar el tiempo de tareas ejecutadas por operantes
- Control de calidad de productos
- Inspeccionar productos u objetos sin contacto físico
- Reducir el tiempo en procesos

### 1.2.1 Aplicaciones y métodos de visión artificial

Los patrones deben ser verificados mediante un control de precisión establecido para la selección de objetos, creando un sistema exacto para no tener errores ni variabilidades en la comparación y extracción de diversos objetos. (Cognex. 2016)

#### Aplicaciones:

- Identificación de objetos
- Ubicación de la posición de objetos
- Determinación de relaciones entre objetos
- Establecer coordenadas entre varios objetos
- Realización de mediciones
- Inspección de varios objetos
- Mediciones tridimensionales

Los resultados deben ser precisos y fiables, el sistema de localización de objetos debe comparar de manera rápida la forma de los objetos en líneas de producción. Las cuatro categorías de localización deben estar completamente perfeccionadas para lograr un óptimo resultado. (Cognex. 2016)



**Figura 3-1:** Categorías principales de visión artificial

Realizado por: Ludeña, J. 2018

**Guiado:** Permite optimizar recursos en tareas y disminuir operarios, localiza la posición de un objeto específico y compara con otro para garantizar el montaje correcto. Se aplica el guiado para localizar un objeto o pieza en un espacio 2D o 3D. (Cognex. 2016)

Se aplica para el alineamiento y posicionamiento automático de objetos o piezas, esto permite localizar puntos específicos para ensamble de varias piezas.

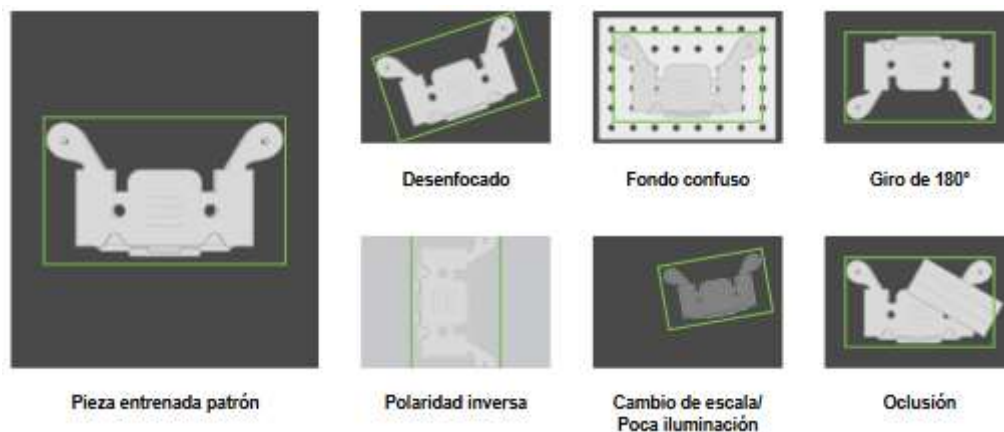




**Figura 4-1:** Imágenes usadas en el guiado

Fuente: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

En algunas ocasiones el guiado sigue patrones geométricos sin mayor variación en su forma, con muy poco contraste e iluminación, así como cambios de rotación. Esto permite que la información localizada mantenga la misma línea de localización. (Cognex. 2016)



**Figura 5-1:** Imágenes guiadas por patrones geométricos

Fuente: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

**Identificación:** El sistema de identificación permite leer códigos de barras, matrices de datos, caracteres y marcajes estos pueden ser alfanuméricos. Con el sistema de identificación se puede reconocer objetos por su tamaño, color y forma. (Cognex. 2016)



**Figura 6-1:** Identificación mediante visión artificial

Fuente: [www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction to Machine Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction to Machine Vision.pdf)

Las industrias mayoristas y minoristas emplean este tipo de visión artificial, pero en algunos casos se ha implementado los códigos 2-D como matrices, esto permite tener mayor información sobre los productos.

**Medición:** Consiente en calcular la distancia requerida entre dos o más objetos si las medidas no son correctas el sistema envía una señal de error en el control esto provoca que el objeto sea rechazado. (Cognex. 2016)

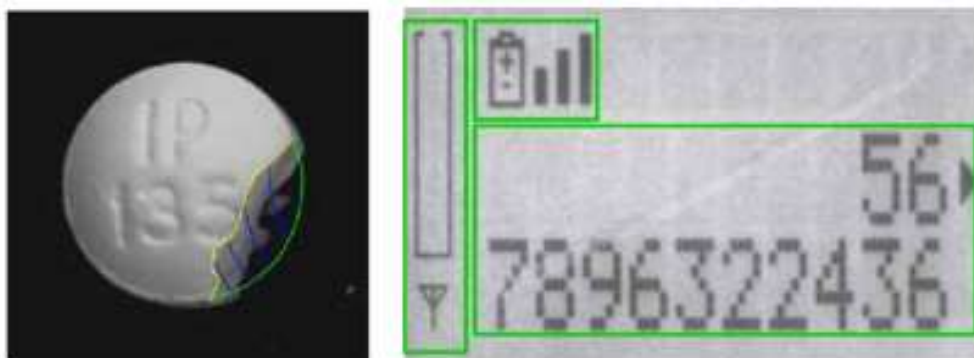


**Figura 7-1:** Sistema de medición con visión artificial

**Fuente:** [www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction to Machine Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

Cuando los objetos entran al campo de visión se calcula la distancia entre los puntos de la imagen manteniendo un rango hasta 0.0254 milímetros, esto permite facilitar y optimizar el trabajo en las industrias.

**Inspección:** Permite detectar imperfecciones, defectos y otras fallas que pueda afectar a un producto u objeto manufacturado. Este sistema es muy aplicado en el área de medicina su principal función es verificar la integridad de los productos. (Cognex. 2016)



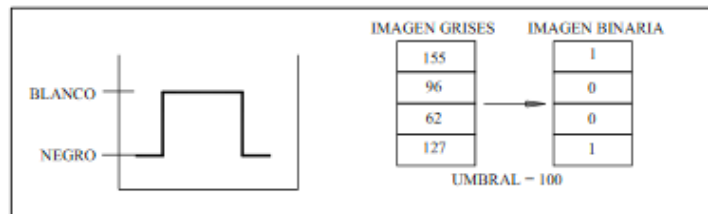
**Figura 8-1:** Sistema de inspección por visión artificial

**Fuente:** [www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction to Machine Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

### Métodos:

- **Digital:** El resultado obtenido tras el muestreo especifica el valor en números positivos estableciendo a cada valor un pixel de brillo que representa a la imagen original.

- **Pixel:** Se considera pixel a cada cuadro que componen una imagen digital, la resolución estándar es de 512 x 484 pixel.
- **Nivel de gris:** La intensidad de brillo es cuantificada al momento de procesar una imagen digital provocando un cambio en su tono original y convirtiendo en escala de grises.
- **Imagen binaria:** Son imágenes que poseen únicamente dos niveles de color (negro y blanco), se conoce también como nivel UMBRAL.



**Figura 9-1:** Esquema de imagen binaria

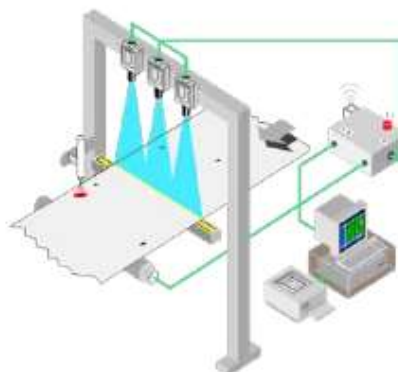
**Fuente:** <http://www.etitudela.com/celula/downloads/visionartificial.pdf>

- **Escena:** Emplea todos los elementos que utiliza la composición del espacio de un objeto como: cámara utilizada, datos ingresados, patrón de imágenes y entrada y salida de control de calidad.
- **Ventana de medida (Windows):** Recoge el área específica en la cual va hacer analizada la imagen.

### 1.3 Sistemas de visión artificial

Los sistemas de visión artificial se clasifican en tres tipos: 1D, 2D y 3D.

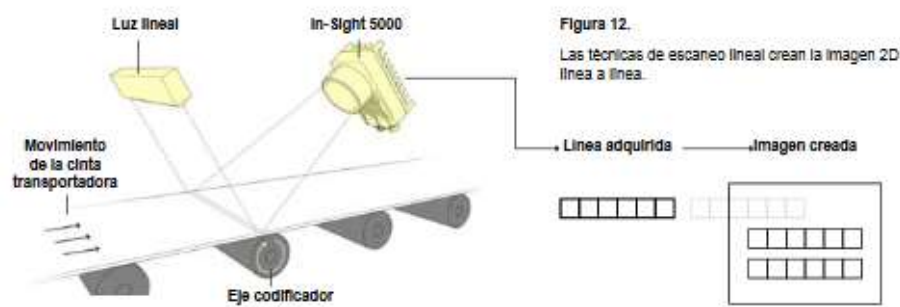
- **Sistemas de visión 1D:** Este tipo de sistema es guiado por una señal digital que sigue líneas en lugar de imágenes, esto permite detectar y clasificar desperfectos en materiales o productos de una producción en serie. (Cognex, 2016)



**Figura 10-1:** Sistema de visión 1D

**Fuente:** [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

- **Sistema de visión 2D:** Emplea dos métodos de escaneos, lineal y superficial los mismos generan imágenes 2D en forma paralela. (Cognex, 2016)



**Figura 11-1: Sistema de visión 2D**

Fuente: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

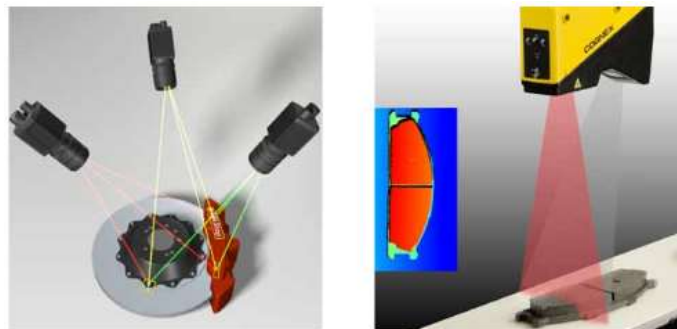
- **Escaneo superficial y escaneo lineal:** Este sistema permite inspeccionar piezas de formas cilíndricas optimizando el tiempo de revisión o escaneo. La ventaja del escaneo lineal es que se adapta a espacios muy estrechos y son adecuados para producción en serie o productos en movimiento constante, en estas circunstancias el escaneo tiene mayor resolución que las cámaras habituales. (Cognex, 2016)



**Figura 12-1: Sistema de visión lineal**

Fuente: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

- **Sistema 3D:** El sistema de visión 3D utiliza varias cámaras y sensores laser, esto permite proporcionar información sobre la orientación, ubicación y posición del objeto. (Cognex, 2016)



**Figura 13-1: Sistema de visión 3D**

Fuente: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

Los sensores laser permiten que las cámaras puedan inspeccionar los objetos calculando la medida y volumen. El objeto o pieza es escaneado completamente generando parámetros de altura con una precisión de 20  $\mu\text{m}$ . (Cognex, 2016)

#### 1.4 Fases del sistema de visión artificial

Es necesario identificar todos los elementos que contienen las imágenes para proceder a descomponer los segmentos que generalmente se dividen en cuatro fases. (Bustamante, S. 2014 pp. 11)

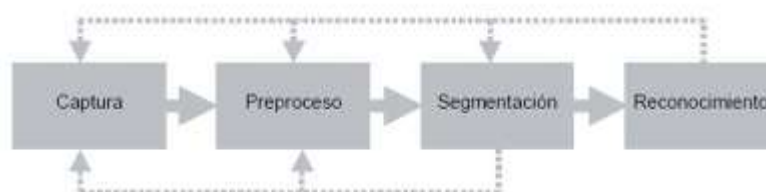
**Tabla 1-1:** Fases del sistema de visión artificial

<b>Primera fase:</b>	<b>Captura:</b> Se denomina etapa sensorial que básicamente consiste en la adquisición o captura de la imagen en su entorno natural.
<b>Segunda fase:</b>	<b>Procesamiento:</b> Se conoce como procesamiento previo, las imágenes pasan por filtros y sufren transformaciones geométricas eliminando partes innecesarias y sobresaliendo las más importantes.
<b>Tercera fase:</b>	<b>Segmentación:</b> Es la fase de segmentación en la cual aísla zonas que realzan la escena.
<b>Cuarta fase:</b>	<b>Reconocimiento:</b> En la última fase las imágenes son reconocidas y clasificadas, se evidencia claramente todos los objetos comprendidos en la escena.

**Realizado por:** Ludeña, J. 2018

**Fuente:** [http://opac.pucv.cl/pucv\\_txt/txt-4500/UCE4968\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-4500/UCE4968_01.pdf)

Si una de las fases falla es necesario hacer el proceso de retroalimentación para evitar errores en la fase final. Es decir, si la fase de procesamiento es errónea se debe regresar a la fase sensorial y después continuar el proceso. (Bustamante, S. 2014 pp. 12)

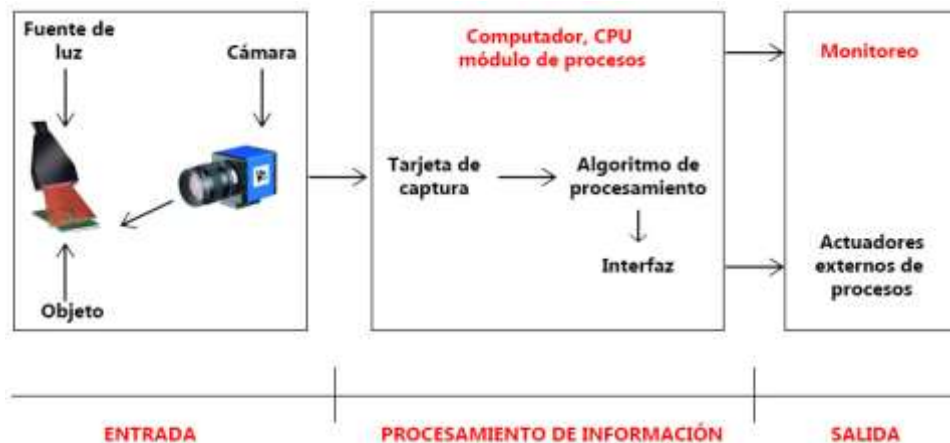


**Figura 14-1:** Fases del sistema de visión artificial

**Fuente:** [http://opac.pucv.cl/pucv\\_txt/txt-4500/UCE4968\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-4500/UCE4968_01.pdf)

## 1.5 Componentes del sistema de procesamiento

Los sistemas de procesamiento y formación de imágenes están constituidos por subsistemas de iluminación, captación de imagen y adquisición de señal que por medio de algoritmos permiten transformar información de alto nivel y ser transmita por medio de imágenes. (Bustamante, S. 2014 pp. 13)



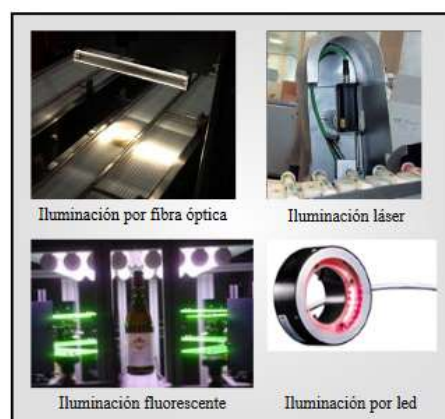
**Figura 15-1:** Procesamiento de imagen

Realizado por: Ludeña, J. 2019

**Subsistema de iluminación:** Conjunto de objetos que producen luz electromagnética como lámparas, filtros de luz, laser entre otros.

La iluminación cumple un papel importante en el proceso de visión artificial, para el desarrollo del sistema el objeto debe ser iluminado en el entorno y bajo estudio. (Vargas, V. 2010 pp. 11)

Existen varios tipos de iluminación que se aplican en visión artificial:



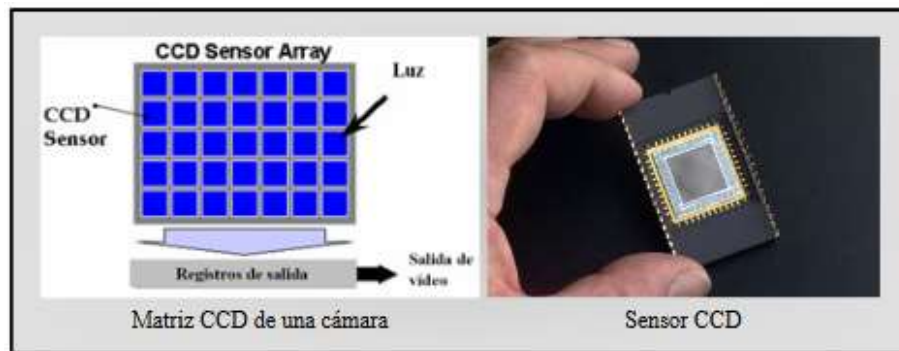
**Figura 16-1:** Tipos de iluminación

Fuente: <https://tesis.ipn.mx/bitstream/handle/123456789/7250/68.pdf?sequence=1&isAllowed=y>

- Por fibra óptica
- Floreciente

- Por Led
- Láser
- Direccional
- De campo
- Lateral
- Difusa
- Coaxial

**Subsistema de captación:** Las cámaras poseen sensores que convierten la radiación luminosa, radiación gama, radiofrecuencia entre otras en señales eléctricas. Las cámaras utilizan un sistema basado en CCD los cuales se dividen en diferentes tipos. (Vargas, V. 2010 pp. 11)



**Figura 17-1: Sensor CCD**

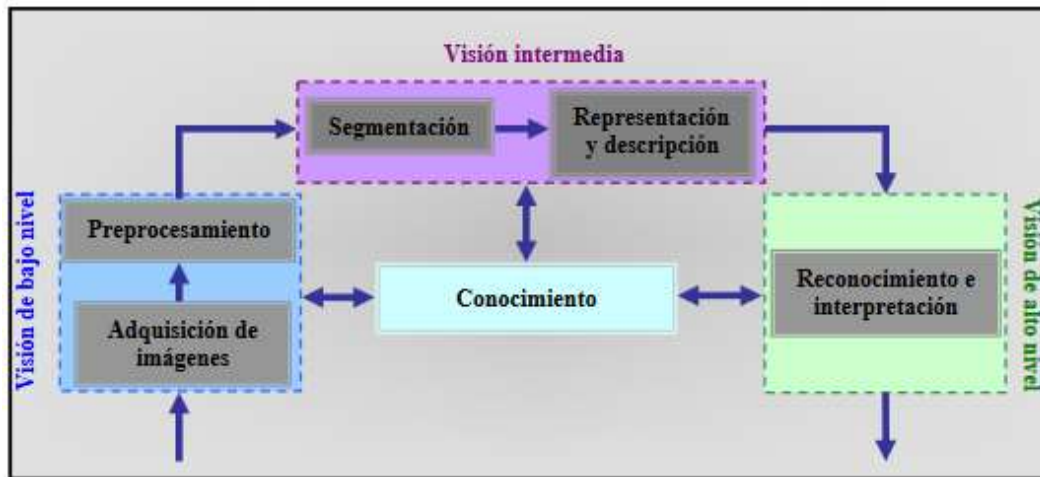
Fuente: <https://tesis.ipn.mx/bitstream/handle/123456789/7250/68.pdf?sequence=1&isAllowed=y>

- **CCD progresivo:** Transmite e integra todos los campos a la vez, realizan barridos progresivos por lo que son ideales para capturar imágenes en movimiento.
- **CCD entrelazado:** Adquiere y transmite todos los campos por separado.
- **CCD digital:** Transmite e integra todos los campos visibles. (Vargas, V. 2010 pp. 12)

**Subsistema de adquisición:** La señal eléctrica es convertida en señal de imagen o video, existen las de tipo digital y analógicas.

**Subsistema de procesamiento:** Los algoritmos obtenidos son tratados en un computador o varios. Las imágenes digitales son procesadas hasta convertirse en información de alto nivel.





**Figura 18-1:** Procesamiento de imágenes

Fuente: <https://tesis.ipn.mx/bitstream/handle/123456789/7250/68.pdf?sequence=1&isAllowed=y>

Las imágenes digitales están compuestas por tres propiedades descritas en la tabla 2-1.

**Tabla 2-1:** Propiedades de las imágenes digitales

PROPIEDADES DE LAS IMÁGENES DIGITALES	
<b>Resolución:</b>	Básicamente es el número de filas y columnas que ocupan los pixeles. Mientras más pixeles poseen las imágenes la resolución es mayor.
<b>Definición:</b>	Muestra el número de colores que posee la imagen. También se lo conoce como profundidad de pixel ya que permite modular los números de bits.
<b>Numero de planos:</b>	Indica el número de arrays de pixel, es decir los planos de la imagen. Cuando una imagen está en escala de grises se encuentra conformada por un solo plano, cuando la imagen esta en tonos RGB está construida por tres planos.

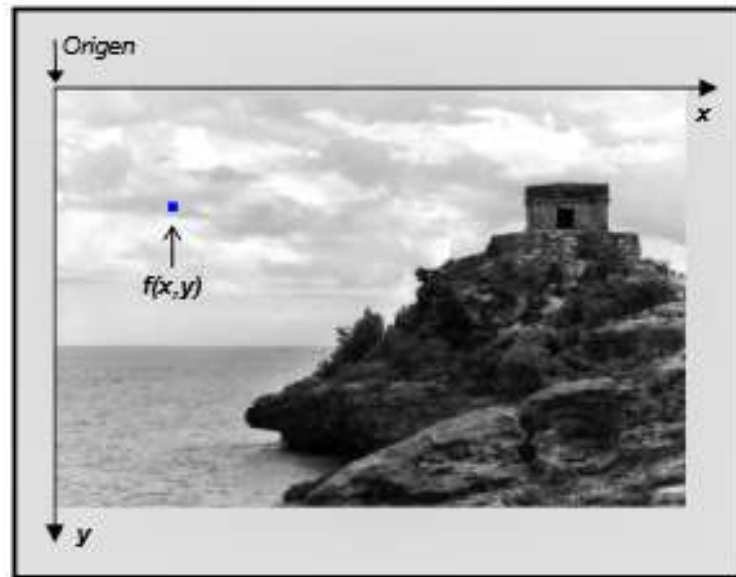
**Realizado por:** Ludeña, J. 2018

Fuente: <https://tesis.ipn.mx/bitstream/handle/123456789/7250/68.pdf?sequence=1&isAllowed=y>

Al procesar imágenes mediante computadoras se puede modificar valores como: cantidad de pixel, tonos, nivel de brillo, balance de blancos entre otros.

Existen imágenes formadas en tonos monocromáticos, es decir constituida a un bit por cada pixel en un solo nivel. Las imágenes en tonos RGB están formadas por 256 niveles. (Vargas, V. 2010 pp. 13)





**Figura 19-1:** Representación de imágenes

**Fuente:** <https://tesis.ipn.mx/bitstream/handle/123456789/7250/68.pdf?sequence=1&isAllowed=y>

**Subsistema de periféricos:** Recepta la información de alto nivel, estos pueden ser monitores de alta representación gráfica.

## 1.6 Hardware utilizado en visión artificial

### 1.6.1 Cámaras

La funcionalidad de las cámaras es procesar una imagen y proyectarla a un sistema electrónico. En visión artificial se requieren cámaras con características que permitan controlar el disparo al capturar las imágenes. Este tipo de cámaras permiten realizar un control completo en sensibilidad ISO, velocidad de obturación, tiempo de apertura, etc.

### 1.6.2 Tipos de cámaras

Según su función se clasifican en:

#### La tecnología del elemento sensor

- **Cámaras de tubo:** Utilizan materiales fotosensibles que capturan la imagen y son procesados por electrones.
- **Cámaras de estado sólido CCD (Charge – Coupled – Device):** Emplean materiales semiconductores fotosensibles.

#### La disposición física

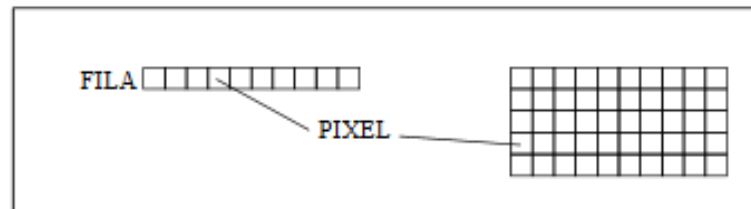
- **Cámaras lineales:** Crean imágenes que van alineadas mediante líneas longitudinales.



**Figura 20-1:** Cámara lineal

**Fuente:** <http://www.etitudela.com/celula/downloads/visionartificial.pdf>

Las cámaras lineales emplean sensores abarcan entre 512 y 8192 píxeles que permite obtener imágenes de alta calidad. Este tipo de cámaras permiten sincronizar los objetos mediante líneas de alta precisión.



**Figura 21-1:** Píxeles en cámaras lineales

**Fuente:** <http://www.etitudela.com/celula/downloads/visionartificial.pdf>

La construcción de imágenes de alta calidad requiere sincronización y alineación precisa, generalmente se utiliza para inspección de objetos como papel, vidrio, telas, etc.

Algunas características principales de estas cámaras son: el número de sensores y píxeles de la unidad, velocidad de lectura de píxeles y tipo de sensores para cada color (RGB).

- **Cámaras matriciales:** Está conformada por una central de píxeles que posicionan de manera muy precisa los sensores en las matrices fotosensibles. Utilizan tecnología CCD que determinan el tamaño de los tubos Vidicón.



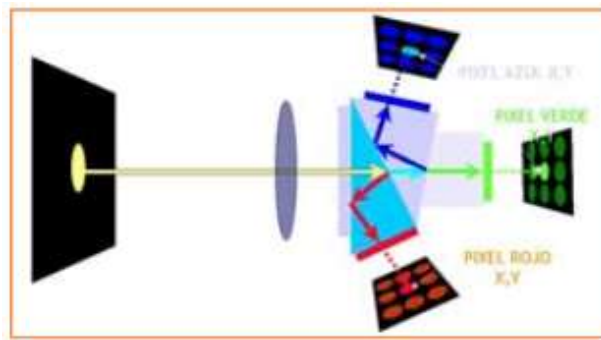
**Figura 22-1:** Cámara matricial

**Fuente:** <http://www.etitudela.com/celula/downloads/visionartificial.pdf>

## Características de los sensores

- **Factor de relleno:** Para que el área de relleno del pixel sea ideal debe ser del 100% de sensibilidad de luz.
- **Transferencia Inter-línea (ITL):** Emplean registros de desplazamiento entre pixeles que permiten transferir y almacenar datos de las imágenes.
- **Transferencia de cuadro:** Almacena áreas de luz que son separadas del área activa esto provoca una pérdida en la velocidad de obturación.
- **Cuadro entero:** Emplean un mecanismo simple de exposición, integración y transporte del factor de relleno.

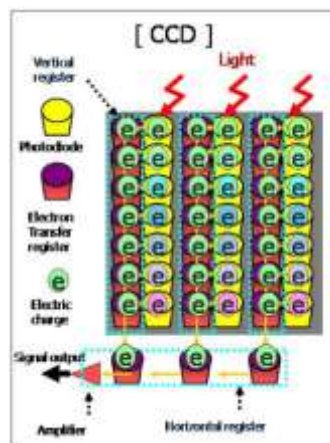
**Cámaras color:** Utilizan métodos más complejos que las cámaras monocromas esto permite obtener mayor cantidad de información.



**Figura 23-1:** Proceso cámara de color

Fuente: <http://www.etitudela.com/celula/downloads/visionartificial.pdf>

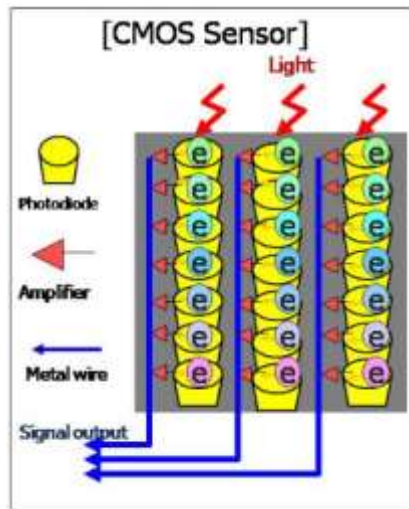
Las cámaras 1CCD emplean sensores con filtros de colores RGB que asimilan el sensor de la vista del ojo humano, este tipo de cámaras transmiten señales analógicas. En cambio, las cámaras 3CCD utilizan tres sensores y un prisma, esto permite que la luz se divida en tres direcciones al momento de llegar al prisma pasando por un filtro RGB y un sensor que procesa la luz.



**Figura 24-1:** Sensor CCD

Fuente: <http://repositorio.uac.edu.co/bitstream/handle/11619/1371/TMCT%200013C.pdf?sequence=1&isAllowed=y>

En el caso de las cámaras con sensores CMOS las celdas son totalmente independientes lo cual produce que cada pixel sea digitalizado en cada celda, esto ayuda favorablemente a reducir costos. (Lozano, G & Orduz J. 2015 pp. 33)



**Figura 25-1: Sensor CMOS**

Fuente: <http://repositorio.uac.edu.co/bitstream/handle/11619/1371/TMCT%200013C.pdf?sequence=1&isAllowed=y>

**ÓPTICAS:** Conducen la luz al sensor de manera moderada para enfocar uno o varios objetos. Para obtener mejores resultados es necesario emplear la siguiente formula:

$$\text{Distancia} = \frac{\text{Tamaño del sensor} \times \text{Distancia al objeto}}{\text{Tamaño del objeto}}$$

Las cámaras ópticas utilizadas especialmente en visión artificial deben cumplir con características específicas que permitan obtener imágenes de calidad y precisión.

#### **Características:**

Iluminación

Sensor de la cámara

Tamaño del objeto

Forma del objeto

Distancia entre objetos

Espacio disponible

### 1.6.3 Raspberry Pi

Es un mini ordenador, esta compuesto por una placa de tamaño reducido, basado en el sistema operativo Linux por lo cual consume pocos recursos. Su función electrónica está incorporada con pines GPIO (General Purpose Input/Output) su comunicación es por medio de UART (Universal Asynchronous Receiver-Transmitter), I<sup>2</sup>C (Inter-Integrated Circuit) y SPI (Serial Peripheral Interface). (Maldonado, L. 2018 pp. 16)



**Figura 26-1:** Raspberry Pi

**Fuente:** <https://uae.souq.com/ae-en/raspberry-pi-3-model-b-10522094/i/>

Hasta la actualidad existen tres modelos de Raspberry Pi, su característica principal es el procesador, la unidad de almacenamiento y el aumento de puertos USB. (González, C. 2015 pp. 25)

A continuación, se describen las características principales de los modelos de Raspberry Pi:

**Tabla 3-1:** Características de los modelos de Raspberry Pi

MODELOS RASPBERRY PI			
Características	Modelo A+	Modelo B+	RPI Modelo B
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM11 ARMv6 700MHz	ARM11 ARMv6 700MHz	ARM11 ARMv7 ARM Cortex-A7 4 núcleos @ 900MHz
GPU	Broadcom VideoCore IV 250MHz OpenGL ES 2.0		
RAM	256 MB LPDDR SDRAM 400 MHz	512 MB LPDDR SDRAM 400 MHz	1 GB LPDDR SDRAM 450 MHz
USB	1	4	4
Salidas de video	HDMI 1.4 @ 1920X1200 px		

Almacenamiento	Micro SD		
Ethernet	No	Si 10/100 Mbps	Si 10/100 Mbps
Tamaño	85,60 x 56,5 mm	85,60 x 56,5 mm	85,60 x 56,5 mm
Peso	23g	45g	45g

**Realizado por:** Ludeña, J. 2018

**Fuente:** <http://bibing.us.es/proyectos/abreproy/90330/fichero/Memoria+TFG.pdf>

#### 1.6.4 Arduino

Todos los tipos de Arduinos son basados en plataformas de código abierto, permite la comunicación del software a un ordenador. Los sensores que permite la recepción de entradas que realizan cambios en el entorno. (Hidalgo, W. 2015 pp. 21)

Existen varios tipos de Arduinos que poseen distintas características, los más importantes están enunciados a continuación:

**Tabla 4-1:** Características de los Arduinos

Especificaciones	Arduino UNO	Arduino MEGA	Arduino Leonardo	Arduino NANO
Micro controlador	ATmega 328	ATmega 2560	ATmega32u4	ATmega 328
Voltaje operativo	5V	5V	5V	5V
Tensión de entrada	7-12V	7-12V	7-12V	7-12V
Voltaje de entrada	6-20V	6-20V	6-20V	6-20V
Pines digitales de entrada y salida	14 (6 son salida PWM)	54 (14 son de salida)	20	22 (6 son para PWM)
Pines análogos de entrada	6	16	7	8
Corriente DC por cada Pin E/S	40mA	40mA	40 mA	40 mA
Corriente DC entregada	50mA	50mA	50mA	19mA

Capacidad de memoria	32 KB (ATmega328) 0.5 KB son utilizados para el arranque	256 KB (8KB son usados por el bootloader)	32 KB (4 KB son usados por el bootloader)	32 KB (2 KB son usados por bootloader)
SRAM	2 KB	8KB	2.5 KB	2 KB
EEPROM	1 KB	4KB	1 KB	1 KB
Clock Speed	16MHz	16MHz	16MHz	16MHz

**Elaborado por:** Ludeña, J. 2019

**Fuente:** <https://www.electrontools.com/Home/WP/2018/06/19/arduino-mega-2560-caracteristicas/>

## 1.7 Software de programación

### 1.7.1 Python

Según David Masip Rodó, Python es un lenguaje de programación libre basada en IronPython (codificada en C#) o JPython (codificada en Java), en la actualidad existen múltiples implementaciones en función que construye el intérprete.

Para instalar las versiones de Python es necesario recurrir al sitio web oficial, aquí se encuentra toda la información para instalar el entorno y los paquetes para sistemas Linux, Windows o MacOS. (Masip, D. 2017)

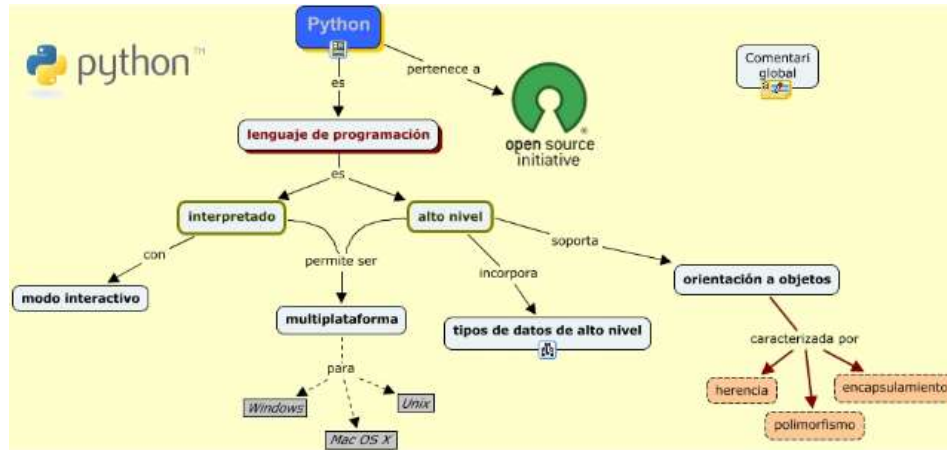
```
macbook:~ david$ python
Python 2.6.6 (r266:84292, Oct 16 2010, 21:41:03)
[GCC 4.0.1 (Apple Inc. build 5490)] on darwin
Type 'help', 'copyright', 'credits' or 'license' for
more information.
>>> print 'Hello World'
Hello World
>>> quit()
macbook:~ david$
```

**Figura 27-1:** Ejemplo de lenguaje Python

**Fuente:** [https://www.exabyteinformatica.com/uoc/Inteligencia\\_artificial/Inteligencia\\_artificial\\_avanzada/Inteligencia\\_artificial\\_avanzada\\_\(Modulo\\_2\).pdf](https://www.exabyteinformatica.com/uoc/Inteligencia_artificial/Inteligencia_artificial_avanzada/Inteligencia_artificial_avanzada_(Modulo_2).pdf)

### 1.7.1.1 Características

El lenguaje Python posee características que permiten crear diferentes tipos de programas, se descompone en sentencias, expresiones, objetos y módulos.



**Figura 28-1:** Características de Python

Fuente: <http://thegangstersnsk.blogspot.com/2015/03/lenguaje-de-programacion-python.html>

- **Interpretado:** No compila el código antes de ser ejecutado.
- **Modo interactivo:** La sentencia se ejecuta y el resultado se visualiza, esto facilita al programador a verificar el código.
- **Multiplataforma:** Desarrollado para sistemas informáticos.
- **Orientado a objetos:** Crea programas con códigos reutilizables.

### 1.7.2 OpenCv

La librería OpenCv (*Open Source Computer Vision*) se emplea para el procesamiento de imágenes en visión artificial, se emplea también en sistemas de seguridad, programas de detección de movimiento, reconocimiento de objetos, control de procesos, robótica entre otros. (Viera, G. 2017)

Se considera una librería gratuita y libre escrita en C y C++, se emplea con cualquier tipo de procesador y funcionan con sistemas como Windows, Linux, Android, Mac OSX e iOS, por lo cual permite crear aplicaciones en tiempo real. (Viera, G. 2017)

**Tabla 5-1:** Componentes Principales de OpenCV

CV	Contiene algoritmos que permite el procesamiento de imágenes y visión por ordenador
MI	Posee herramientas de agrupación y clasificadores estadísticos.



HighGUI	Realiza funciones de entrada y salida (I/O) que permite subir y almacenar imágenes o videos.
CXCore	Contiene algoritmos con funciones gráficas. Permite recibir información de CV, MIL Y HighGUI.
CvAux	Posee algoritmos experimentales.

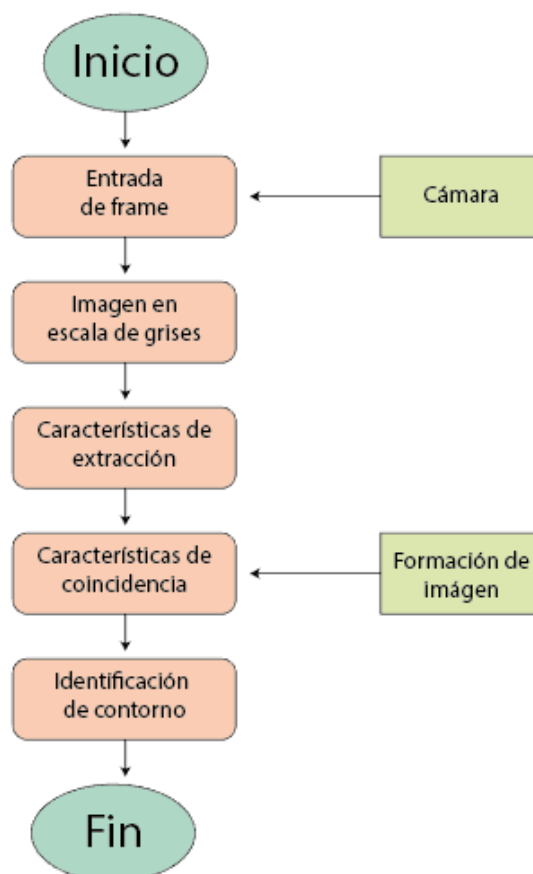
**Realizado por:** Ludeña, J. 2018

**Fuente:** [https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME\\_218.pdf?sequence=1](https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME_218.pdf?sequence=1)

La biblioteca OpenCV utiliza funciones que permite identificar personas u objetos, inspección de objetos, reconocimiento de rostro, reconstrucción 3D entre otros, esto debido a la infraestructura accesible sobre el procesamiento de imágenes. (Viera, G. 2017)

### 1.7.3 SIFT & FLANN

Los algoritmos SIFT & FLANN se pueden realizar en Matlab, Python y C.



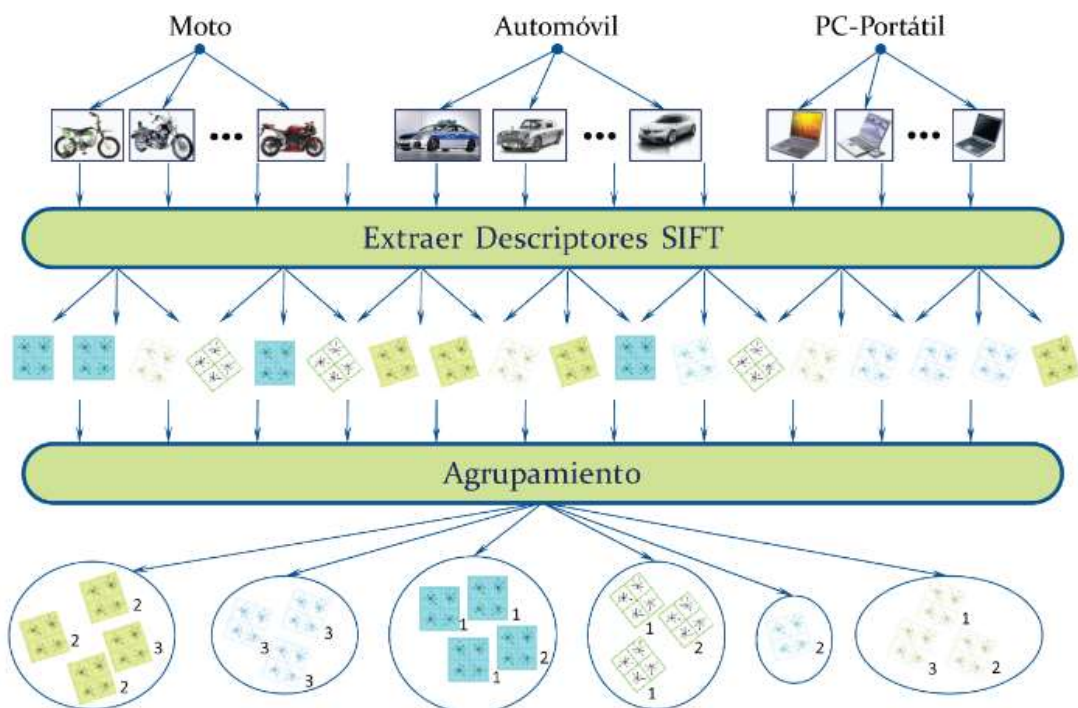
**Figura 29-1:** Algoritmo SIFT & FLANN para detección de objetos

**Realizado por:** Ludeña, J. 2019

La función básica de FLANN es realizar búsquedas cercanas de píxeles en espacios de gran dimensión esto permite crear sistemas de capturas de imágenes por contornos en distintos campos. (Kannan et al. 2018 pp. 2605)

El algoritmo SIFT desarrolla el método de coincidencia de imágenes por detección de esquinas, existen distintas aplicaciones para este algoritmo entre una de ellas es el reconocimiento facial. (Kannan et al. 2018 pp. 2605)

Los descriptores del algoritmo SIFT busca las coordenadas en puntos establecidos para direccionarlos a una orientación determinada, cada descriptor debe ser diferente entre ellos. Poseen histogramas que calculan valores de magnitud en dimensiones de 16 x 16 píxeles de 8 bits por cada uno. (Duarte, M & Chang, L. 2016 pp. 3)



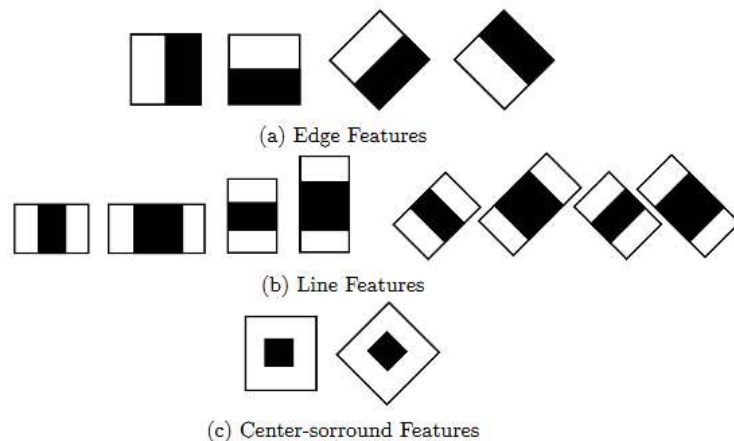
**Figura 30-1:** Clasificación de objetos mediante SIFT

**Fuente:** <http://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/chang-duarte.pdf>

El algoritmo SIFT permite agrupar objetos de forma aglomerada, cada agrupamiento es iniciado por un descriptor el cual selecciona a todos los objetos de un mismo tipo en un solo grupo. La similitud entre objetos se vincula de manera simple, media o completa en el cual se obtiene valores de umbral para detener el proceso. (Duarte, M & Chang, L. 2016 pp. 5)

#### 1.7.4 HARR Cascade

La clasificación de imágenes por medio del algoritmo HAAR Cascade permite la detección de objetos bajo términos de escala y forma, en la etapa de entrenamiento procede a clasificar objetos en diferentes locaciones mediante parámetros de rechazo estadístico. (Reinius, S. 2013 pp. 13)



**Figura 31-1:** Clasificador de patrones mediante HAAR Cascade

**Fuente:** <http://www.diva-portal.org/smash/get/diva2:601707/fulltext01.pdf>

Los principales clasificadores son creados a partir de patrones pequeños los cuales se van acumulando para aumentar muestras clasificadas en las próximas interacciones. Estas acumulaciones crean patrones que calculan sectores que poseen más iluminación y restan sectores oscuros eliminando probabilidades de muestras falsas. (Reinius, S. 2013 pp. 14)

La información anteriormente recopilada se toma como punto de partida para la creación de un sistema de supervisión para niños, empleando los software y hardware necesarios que permitan el desarrollo del mismo.

## CAPITULO II

### 2. MARCO METODOLÓGICO

Para la implementación de un sistema de supervisión para niños de 2 a 4 años mediante visión artificial es indispensable seleccionar los tipos de hardware y software más apropiados que contribuyan con el perfecto desarrollo del sistema.

#### 2.1 Selección de Dispositivos

Para la selección de dispositivos es necesario tomar en cuenta las características principales de cada elemento, el objetivo principal es cumplir con la implementación del sistema de seguridad para supervisión de niños entre 2 a 4 años.

#### 2.2 Hardware

##### 2.2.1 Raspberry Pi 3

Permite al usuario emplearla en diferentes usos, posee software y hardware libre, dispone de sistema operativo Raspbian 4.14 basado en Linux que permite controlar varios dispositivos, se conecta mediante HDMI hacia la pantalla, se alimenta mediante una fuente de 5V y 3A el conector es micro USB o tipo V8.



**Figura 1-2:** Placa Raspberry Pi 3 Modelo B  
Realizado por: Ludeña, J. 2019

Al conectar el mouse como dispositivo de salida facilita el uso de la interfaz, es decir permite el control de la función de los botones. Posee ranuras que reconocen dispositivos de almacenamiento externo, la entrada Ethernet facilita la conexión a la red. Se conecta al Arduino UNO por puerto UART mediante conexión serial.

Se elige este tipo de Raspberry principalmente por el bajo costo, gran utilidad y la proyección para elaboración de proyectos o programas basados en software libre como por ejemplo el uso de Python el mismo que se utiliza en el desarrollo del sistema.

### 2.2.2 Arduino Uno



**Figura 2-2:** Arduino Uno  
Realizado por: Ludeña, J. 2019

Es un micro controlador (datasheet) basado en ATmega328P. Los pines de alimentación funcionan mediante conexión USB o por medio de fuente externa. Debido a la variedad y ventajas que muestra, se optó por utilizar la tarjeta Arduino UNO ya que la mayoría de librerías que se emplea para el desarrollo de prototipos están dedicadas y comprobadas en dicha tarjeta como por ejemplo la librería del módulo SIM 900 como se observa en la tabla 1-2, dicha librería únicamente funciona cuando se utiliza con Arduino UNO por las configuraciones internas que necesita el módulo.

Se encarga de controlar todos los dispositivos de salida como son el control de voz el control de la baliza y el sistema de advertencia por GSM, se implementó de esta manera para que el procesamiento de la Raspberry Pi sea más rápida.

**Tabla 1-2:** Conexión de pines del Arduino UNO

Pines de Arduino UNO	Componente	Descripción
0	Sin conexión	
1	Sin conexión	
2	Relé pin de activación	Controla la salida para el módulo Relay para activación de la alarma
3	Sin conexión	
4	Sin conexión	
5	Sin conexión	

6	Sin conexión	
7	Tx SIM 900	Transmisión serial, los datos que recibe la Arduino de la SIM 900
8	Rx SIM 900	Transmisión de datos del Arduino con la SIM 9000
9	Enable SIM 900	Controla la activación y desactivación de la SIM 900
10	Tx DF Player	Transmite de manera serial del archivo mp3 que se va a reproducir
11	Rx DF Player	Pin de recepción de datos que recibe el Arduino desde el DF Player
12	Sin conexión	
13	Sin conexión	

**Realizado por:** Ludeña, J. 2019

### 2.2.3 Módulo GSM SIM 900



**Figura 3-2:** Módulo GSM SIM900

**Realizado por:** Ludeña, J. 2019

Los módulos GSM se denominan estándar ya que su comunicación es totalmente digital, en Europa los módulos GSM usan frecuencia de 900MHz y 1800MHz, en cambio en EEUU emplean frecuencia de 1900MHz. Esto permite transmitir datos digitales, de voz y mensajes multimedia. El módulo SIM900 posee características que permite la comunicación en forma remota. A continuación, se describen en la tabla 1-2.

**Tabla 2-2:** Características de módulo GSM SIM900

Chip SIM900	Posee datasheet con funcionalidad de pin out del chip.
Antena	La antena se conecta mediante un zócalo para cable coaxial.
Zócalo tarjeta SIM	Tiene un zócalo en la parte inferior en la cual debe ir conectada la tarjeta SIM.
Conector de alimentación Jack	La tarjeta se alimenta con 450 mA, pero en algunas ocasiones puede haber variaciones hasta 2 A.
UART	La comunicación en serie es asincrónica mediante hardware de pines D0 y D8 o por medio de software por los pines D7 y D8.
Pines de entrada y salida	Los pines del módulo permiten la conexión con Arduino. Posee pines digitales, analógicos y PWM.
Conectores para micrófono y auricular	Cuenta con conectores tipo mini jack.
Leds indicadores	Muestran el estado de los elementos.

**Realizado por:** Ludeña, J. 2019

**Fuente:** <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/65345/6/radriandTFG0617memoria.pdf>

Se empleó el módulo SIM 900 debido a su composición electrónica ya que debido a las características internas que posee la tarjeta se podría decir que es completa y no necesita módulos de alimentación externa como otros módulos, por ejemplo, la SIM 800 que es necesario adicionar un control de alimentación para su utilización.

#### **2.2.4 Cámara Logitech C270**



**Figura 4-2:** Cámara Logitech C270

**Realizado por:** Ludeña, J. 2019

Permite realizar videos en alta resolución hasta 1280 x 720 pixeles, fotografías de 3.0 megapíxeles, posee micrófono incorporado para videoconferencias, se seleccionó este dispositivo principalmente por su calidad y bajo costo en comparación con una cámara exclusiva para Raspberry.

Se encuentra ubicada en la parte superior del prototipo, básicamente la función que cumple la cámara es capturar las imágenes o videos de los objetos peligrosos para que ingrese en el algoritmo de detección y se realice el entrenamiento. Las principales características se describen a continuación.

**Tabla 3-2:** Especificaciones técnicas de la cámara Logitech C270

Resolución máxima	720p / 30 fps
Enfoque	Foco fijo
Lente	Tecnología estándar
Micrófono	Mono (integrado)
Campo visual	60°
Soporte	Sujetador universal para monitores, portátiles o pantallas LCD
Longitud de cable	1.5 m

**Realizado por:** Ludeña, J. 2019

**Fuente:** <https://www.logitech.com/es-mx/product/hd-webcam-c270>

### 2.2.5 Convertidor DC-DC Step Down



**Figura 5-2:** Convertidor DC-DC Step Down

**Realizado por:** Ludeña, J. 2019

Es una fuente conmutada que permite regular el voltaje de entrada para devolver un voltaje menor de salida, la fuente de alimentación es directa por lo cual la cantidad de calor que transmite es



menor. Se emplea para adecuar el voltaje de alimentación en el circuito. Como se observa en la tabla 3-2 los cambios bruscos de voltaje no son un problema para esta fuente, el suministro de corriente de 5 amperios que usa la tarjeta de alimentación permite conectar todos los componentes de control como Arduino, DFM player entre otros.

Posee las siguientes características:

**Tabla 4-2:** Características del Convertidor DC-DC Setp Down Kim 0551

Convertidor buck	~93%
Voltaje de entrada	12 V
Voltaje de salida	5 V
Voltaje de salida	Fija
Corriente de salida (promedio)	5 A
Corriente de salida (pico)	8 A
Led	Indicador de encendido
Capacitores	470 uF 270 uF

**Realizado por:** Ludeña, J. 2019

**Fuente:** <https://www.geekfactory.mx/tienda/energia-fuentes-de-alimentacion-baterias/convertidor-dc-dc-step-down-buck/>

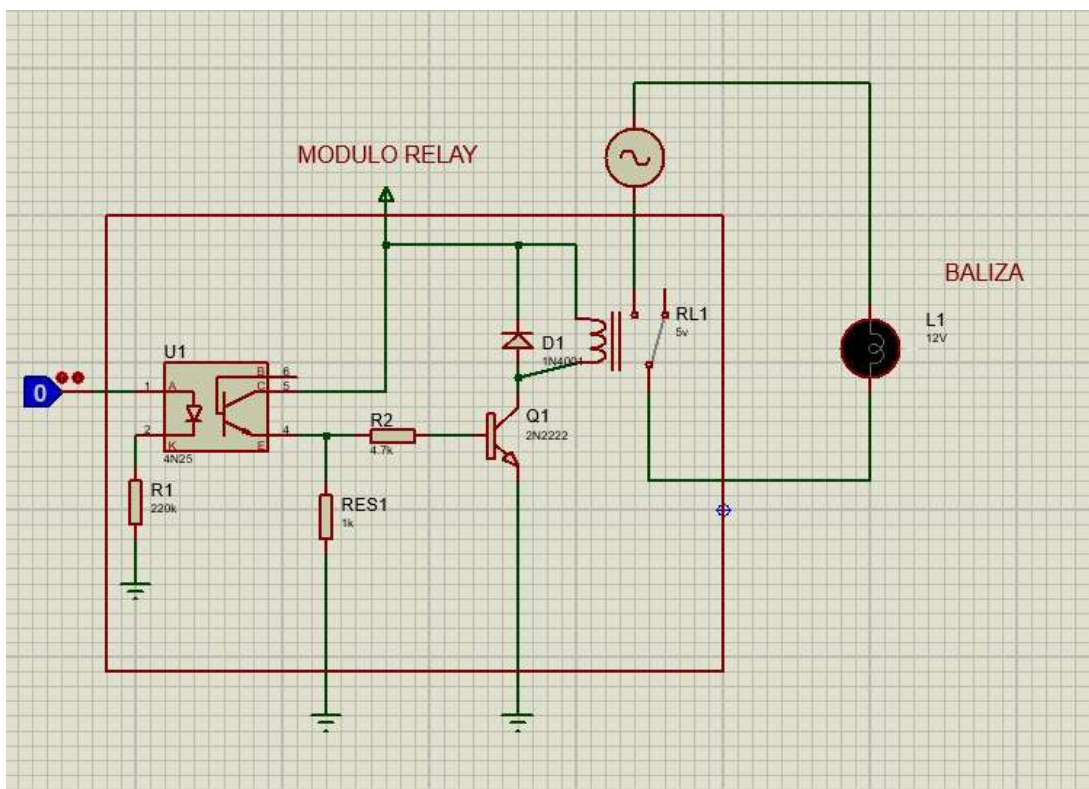
### 2.2.6 Módulo Relé de un canal



**Figura 6-2:** Módulo Relé de un canal

**Realizado por:** Ludeña, J. 2019

Controla el encendido y apagado (switch de carga) opera cargas hasta 250V/10A, al acoplar el relé con el Arduino y el convertidor Dc-Dc Step Down se activa en negativo esto provoca que envíe señal hacia la alarma. (Ver especificaciones técnicas en la tabla 4-2)



**Figura 7-2:** Diagrama de conexión  
Realizado por: Ludeña, J. 2019

**Tabla 5-2:** Especificaciones técnicas del módulo relé de un canal

PARÁMETROS	ESPECIFICACIONES	VALORES UTILIZADOS
Señal de control	TTL (5V o 3.3V)	5V
Voltaje de operación	5V DC	5V
Número de canales	1 CH	1CH
Corriente máxima	10A NA, 5A NC	5A
Capacidad máxima	10A/250VAC – 10A/30VDC	20 A
Activar salida	0 voltios	0
Tiempo de acción	10 ms / 5 ms	10ms

**Realizado por:** Ludeña, J. 2019

**Fuente:** <https://naylampmechatronics.com/drivers/297-modulo-relay-1-canal-5vdc.html>

### 2.2.7 Módulo DF Player Mini



**Figura 8-2:** Módulo DF Player Mini  
Realizado por: Ludeña, J. 2019

Permite reproducir audios grabados, funciona con una MicroSD y soporta formatos MP3, WMA o WAV. Se conecta al parlante en forma directa, se controla mediante comunicación serial por lo cual es 100% compatible con Arduino además se puede implementar botones adicionales para el retroceso y avances de reproducción de audio.

Posee las siguientes características:

**Tabla 6-2:** Características del módulo DF Player Mini

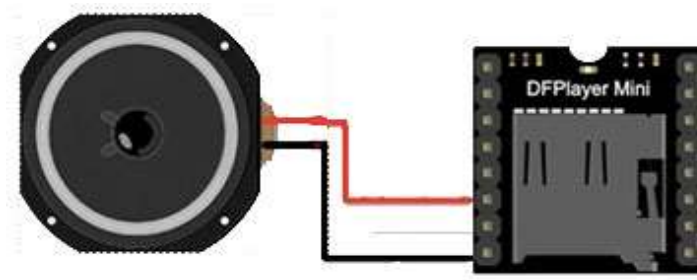
PARÁMETROS	CARACTERÍSTICAS	VALORES USADOS
Voltaje de alimentación	3.2Vdc a 5Vdc	5VDC
Potencia máxima del parlante	3W	3W
Frecuencia soportada en KHz	8 / 11.025 / 12 / 16 / 22.05 / 24 / 32 / 44.1 / 48	8
Tipo de archivos soportado	FAT16 / FAT32 / MP3	MP3
Capacidad máxima de ranura Micro SD	32GB	16 GB
Niveles de volumen	30	20
Puertos de comunicación	1 UART, 1 USB	UART

Salida de audio	Salida para parlante / puerto estero para audífonos	Salida de parlante
Pin	Pin digital para avanzar entre pistas / Pin digital para retroceder	6 pines

**Realizado por:** Ludeña, J. 2019

**Fuente:** <http://www.electronicaplugandplay.com/modulos-plug-and-play/product/520-mod-dfplayer-mini>

### 2.2.8 Parlantes



**Figura 9-2:** Conexión del parlante al DF Player

**Realizado por:** Ludeña, J. 2019

La versión 2.0 se conecta de forma directa a la placa permitiendo reproducir audios MP3. Son usados exclusivamente por el módulo DF Player por lo cual es usado solo por el Arduino y reproducirá las esperas y la acción de los botones del sistema.

### 2.2.9 Baliza



**Figura 10-2:** Baliza

**Realizado por:** Ludeña, J. 2019

Es un elemento de gran impacto visual, se utilizó la baliza porque indica una situación de peligro. Al momento que se envía la señal de alarma se activa este elemento emitiendo una señal luminosa conjuntamente con el módulo GSM.

#### **2.2.10 Pantalla**



**Figura 11-2: Pantalla**  
**Realizado por:** Ludeña, J. 2019

Existe un sin número de pantallas que son compatibles con Raspberry Pi, para la implementación del sistema de supervisión se utiliza la pantalla genérica con entrada a HDMI ya que es de bajo costo en comparación con una pantalla HDMI propia para Raspberry.

### **2.3 Software**

#### **2.3.1 Open CV Library**



**Figura 12-2: Identificador OpenCV**  
**Fuente:** <http://opencv.org/>, 2019

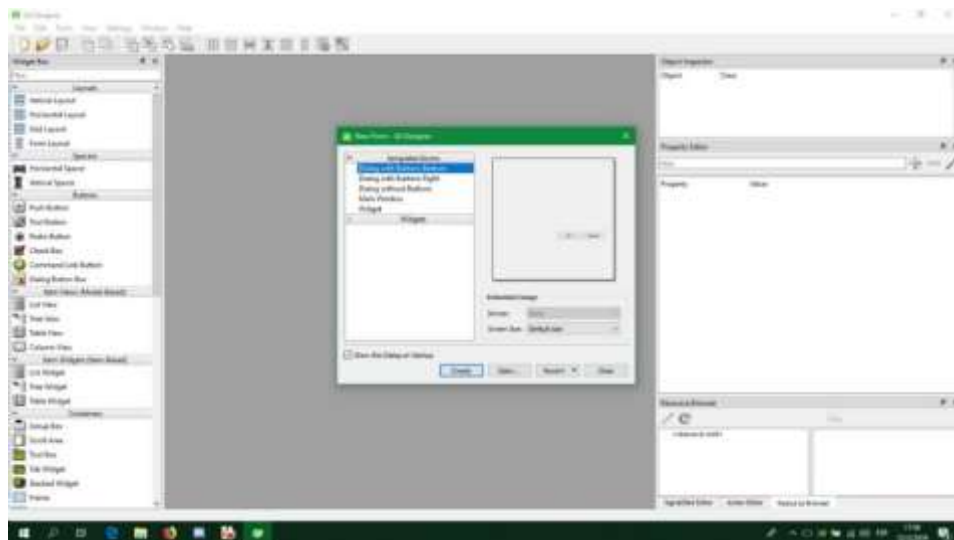
Es una librería con grandes ventajas exclusivamente para el desarrollo de visión artificial. Se utilizará Open CV y su herramienta de reconocimiento de objetos mediante HAAR Cascada ya que es una de las formas más comprobadas y efectivas de reconocimiento.

El HAAR Cascada comparado con otros métodos actuales de reconocimiento tienen un menor consumo de recursos al momento del entrenamiento (hablando exclusivamente de consumo de

GPU) gracias a esto el entrenamiento se puede realizar en casi cualquier dispositivo que tenga sistema operativo Windows o Linux además existen métodos o formas para realizar el entrenamiento dentro del HAAR Cascada, es decir, el entrenamiento en la Guide Training Cascada facilita el desarrollo de este sistema.

### 2.3.2 Qt Designer

Es una biblioteca la cual crea interfaces graficas con Qt Widgets. Personaliza cuadros de diálogos y ventanas, emplea diferentes estilos y resoluciones que permite la reproducción inmediata es decir se visualiza a medida que se va realizando. Esta herramienta facilita la creación de Guides o presentaciones visuales para Python por lo cual se crea una Guide exclusivamente para el sistema para ello se utiliza labels, botones, widget y check box como elementos principales.



**Figura 13-2:** Interfaz Qt Designer  
Realizado por: Ludeña, J. 2019

## 2.4 Descripción del proceso



**Figura 14-2:** Proceso de supervisión  
Realizado por: Ludeña, J. 2019

El sistema de seguridad está elaborado con elementos que permiten la creación del proceso de supervisión artificial como son: cámara, Raspberry Pi, Arduino, Módulo GSM, etc. Básicamente el sistema realizara el siguiente proceso: cuando un objeto peligroso que este dentro de la base de datos del sistema aparezca en pantalla lo detectara inmediatamente pero únicamente se activara cuando dicho elemento sea alzado o levantado por el infante activando los dispositivos de emergencia (baliza y módulo GSM) instalados en el microcontrolador.

El sistema se podrá suspender o habilitar mediante la una interfaz gráfica que está a disposición del usuario.



## 2.5 Desarrollo de la interfaz gráfica


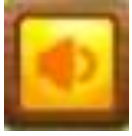




**Figura 15-2:** Diagrama de la interfaz gráfica  
Realizado por: Ludeña, J. 2019

Está controlada por botones cada uno cumple una acción independiente como se observa en la figura 14-2. Las funciones se describen en la tabla 6-2.

**Tabla 7-2:** Descripción de botones empleados en la interfaz gráfica

Icono	Acción	Descripción
	Start cámara	Inicia la cámara para la visualización en un label predeterminado
	Pausa	Pausa la transmisión de la cámara

	Detección	Activa el sistema de detección de objetos
	Activación	Activación del sistema de detección de objetos
	Reinicio	Reinicia el sistema de alarma
	Probar GSM	Sirve para probar la alarma GSM

Realizado por: Ludeña, J. 2019

Cada botón tiene su propia función dentro de la programación en Python, dicha interfaz da como resultado un archivo .ui que en su interior se representa como programación en HTML como se observa en la figura 15-2, por tanto, usar QT Designer facilita la creación de interfaces comparándolos con otras librerías que existen en Python los cuales se programan en el propio código de Python.

```

</widget>
<widget class="QPushButton" name="startVideo">
  <property name="geometry">
    <rect>
      <x>100</x>
      <y>160</y>
      <width>61</width>
      <height>61</height>
    </rect>
  </property>
  <property name="cursor">
    <cursorShape>ArrowCursor</cursorShape>
  </property>
  <property name="autoFillBackground">
    <bool>>false</bool>
  </property>
  <property name="text">
    <string notr="true"/>
  </property>
  <property name="icon">
    <iconset>
      <normaloff>imagenesQt/play.png</normaloff>imagenesQt/play.png</iconset>
    </property>
    <property name="iconSize">
      <size>
        <width>65</width>
        <height>200</height>
      </size>
    </property>
    <property name="shortcut">
      <string/>
    </property>
  </widget>

```

**Figura 16-2:** Código HTML de los botones de la interfaz gráfica

Realizado por: Ludeña, J. 2019



Para describir otro elemento principal dentro de la interfaz se puede denotar la utilización de labels para la representación de gráficas y para mostrar el video desde Python como se puede apreciar en la figura 15-2. La programación del label principal donde se encuentra la imagen de fondo es la misma que los botones, claro está que solo cambia sus propiedades y las dimensiones.

```
</widget>
<widget class="QLabel" name="label_8">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>1181</width>
      <height>531</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>Cooper Black</family>
      <weight>50</weight>
      <bold>false</bold>
    </font>
  </property>
  <property name="text">
    <string/>
  </property>
  <property name="pixmap">
    <pixmap>imagenesQt/fondo.jpg</pixmap>
  </property>
  <property name="scaledContents">
    <bool>true</bool>
  </property>
</widget>
```

**Figura 17-2:** Código HTML de los labels de la interfaz gráfica  
Realizado por: Ludeña, J. 2019

Para unir o comunicar los labels y demás botones con Python es necesario la instalación del Qt Tools en Python para ello se debe descargar dicha librería desde el terminal. La instalación de la librería es relativamente fácil se necesita solo una línea de código para la misma se puede observar en la figura 16-2. Cabe recalcar que toda la parte visual se realizó en Windows ya que es más fácil de realizar, la interfaz terminada se visualiza en la figura 17-2.



**Figura 18-2:** Interfaz gráfica terminada  
Realizado por: Ludeña, J. 2019

### 2.5.1 Entrenamiento de Haar Cascade



**Figura 19-2:** Pantalla principal del Cascade Trainer GUI  
Realizado por: Ludeña, J. 2019

Para el entrenamiento del archivo Cascade se utiliza la guide Cascade Trainer GUI. Facilita y reduce el número de pasos comparado con otras formas de entrenamiento, para esto se cumplen los siguientes pasos:

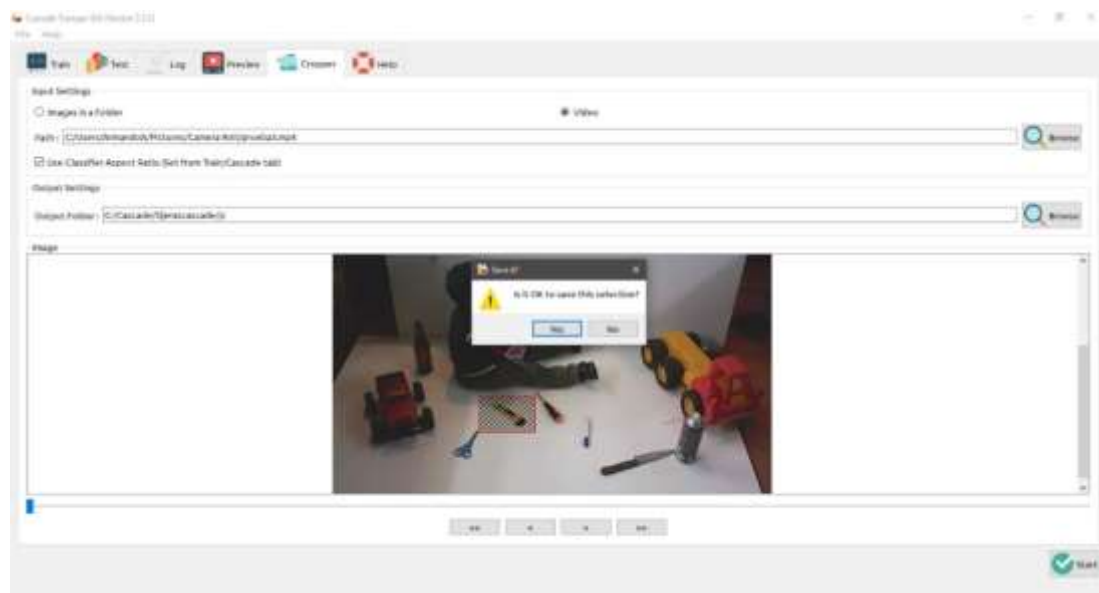
**Primer paso:** Recopilación de información (imágenes o videos), en este caso las imágenes que van hacer reconocidas, pueden ser en forma de video o fotografía para mayor facilidad en la selección de imágenes se prefiere tomar videos de los objetos como se muestra en la figura 22-2.



**Figura 20-2:** Recopilación de imágenes y videos  
Realizado por: Ludeña, J. 2019

**Segundo paso:** Se procede a seleccionar los objetos positivos de las imágenes o videos que se copilaron anteriormente, la aplicación Cascade Trainer GUI además de poseer el entrenador también tiene una aplicación que sirve para seleccionar las imágenes positivas u objetos de los videos o imágenes grabadas. Dicha aplicación se puede usar para imágenes positivas o negativas como se observa en la figura 20-2.

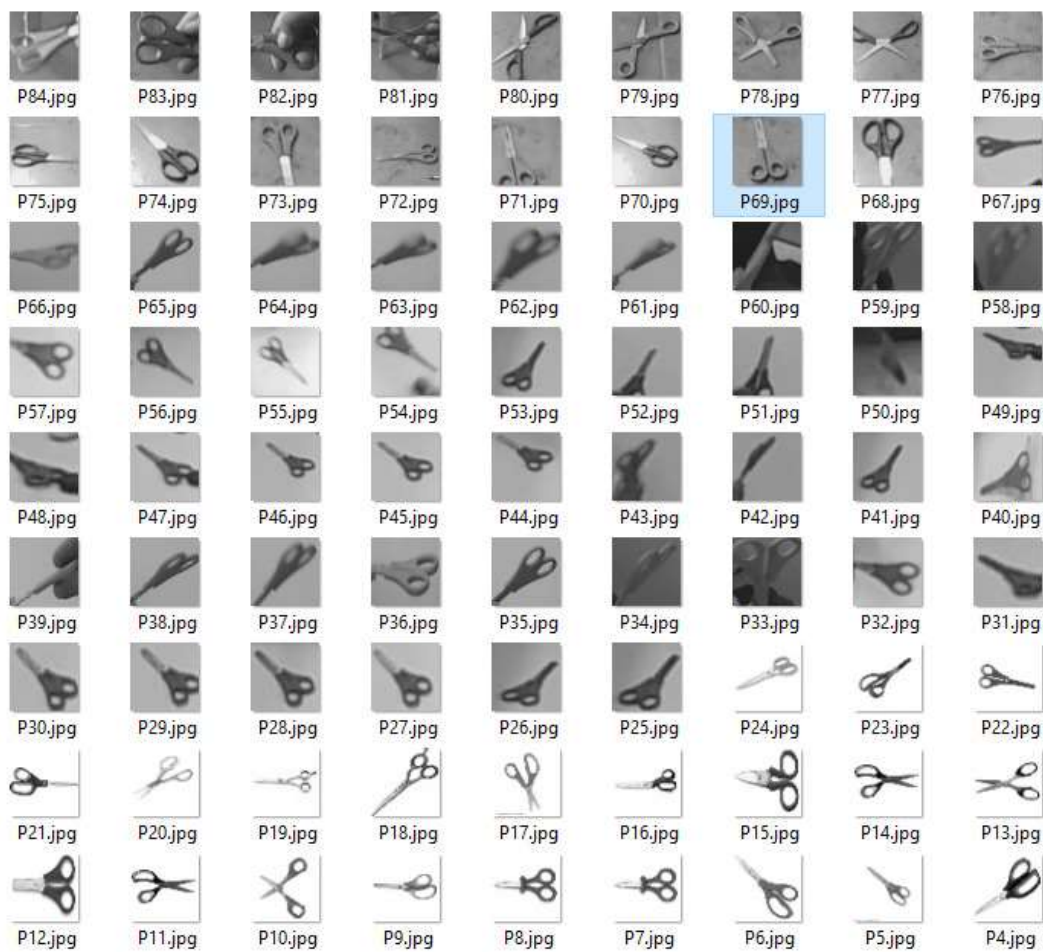
Como se muestra en la figura es necesario seleccionar la imagen positiva o negativa la cual se va a guardar en una carpeta ya antes seleccionada.



**Figura 21-2:** Selección de imágenes positivas  
Realizado por: Ludeña, J. 2019

**Tercer paso:** Después de seleccionar o guardar las imágenes positivas o negativas en las carpetas correspondientes como se muestra en la figura 21-2 se procede a aumentar el número de imágenes

negativas desde el Haar Cascade Master desde la base de datos que ya tiene la aplicación por defecto.



**Figura 22-2:** Imágenes positivas guardadas

Realizado por: Ludeña, J. 2019

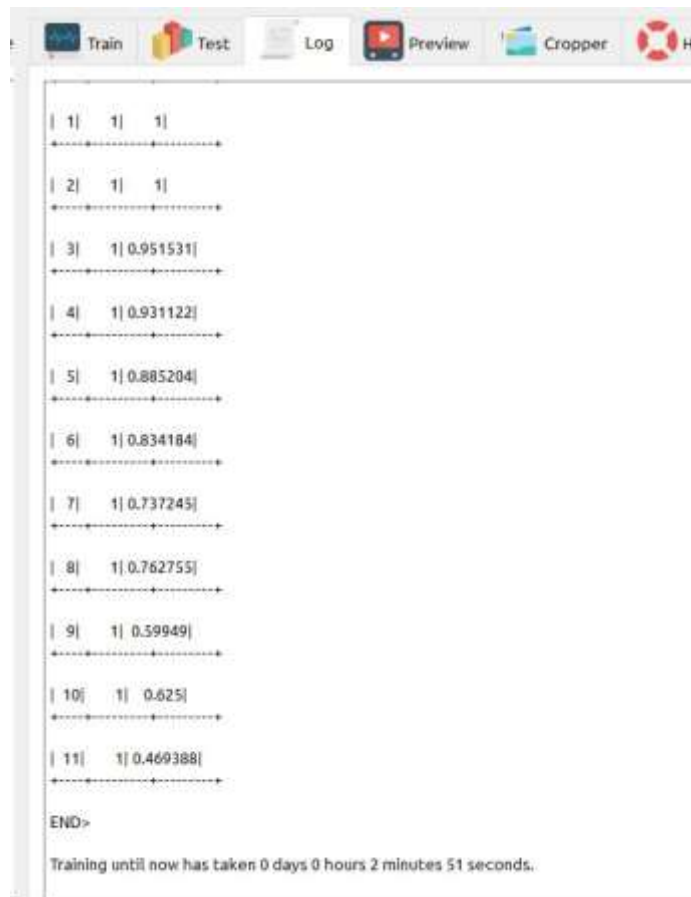
**Cuarto paso:** Se configura el entrenador, por ejemplo, el número de stages, la memoria RAM del entrenador y el número de imágenes positivas y negativas, así como otros parámetros más importantes para el entrenamiento.

Después de configurar se procede a la inicialización, el tiempo del entrenamiento depende del dispositivo que se está utilizando para este caso se empleó una portátil HP Envy 17" con procesador Core i7 y 16 Gb de memoria RAM.



**Figura 23-2:** Configuración del entrenador  
**Realizado por:** Ludeña, J. 2019

**Quinto paso:** Final del entrenamiento, después de un determinado tiempo de entrenamiento dependiendo del número de imágenes que en este caso fue de 350 imágenes positivas y 1700 imágenes negativas aproximadamente tuvo un tiempo de entrenamiento de 5.40 horas al finalizar el Cascade se obtienen Stages que se han probado en el entrenamiento y un archivo XML que se utilizará en la programación.



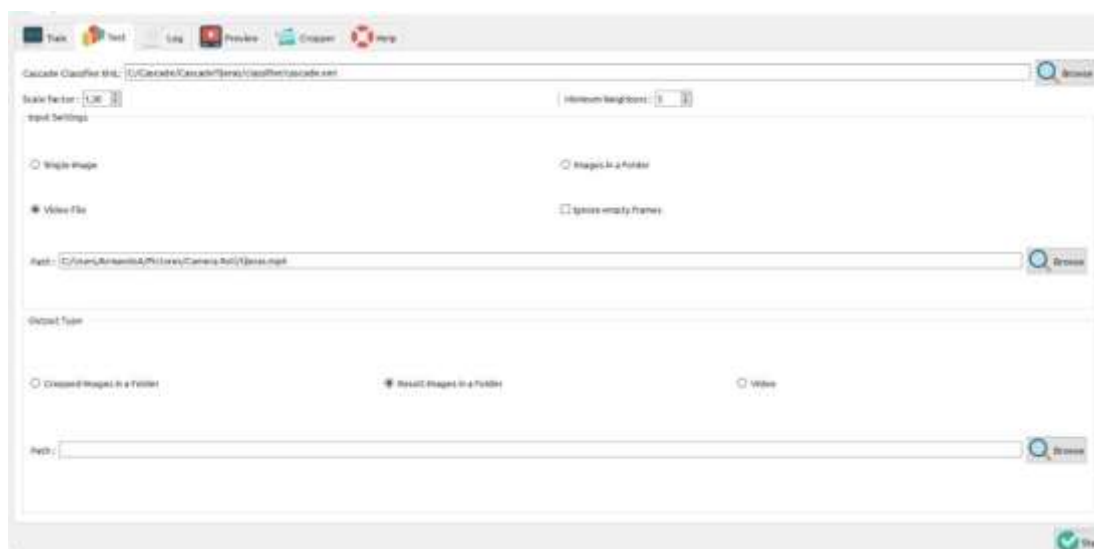
**Figura 24-2:** Final del entrenamiento  
**Realizado por:** Ludeña, J. 2019

NAME	LAST MODIFIED	TYPE	SIZE
<b>cascade.xml</b>	30/5/2019 4:09	Documento XML	49 KB
log.txt	30/5/2019 4:09	Documento de tex...	795 KB
part1.xml	29/5/2019 22:47	Documento XML	1 KB
stage0.xml	29/5/2019 22:47	Documento XML	1 KB
stage1.xml	29/5/2019 22:47	Documento XML	2 KB
stage2.xml	29/5/2019 22:47	Documento XML	2 KB
stage3.xml	29/5/2019 22:48	Documento XML	1 KB
stage4.xml	29/5/2019 22:48	Documento XML	2 KB
stage5.xml	29/5/2019 22:48	Documento XML	2 KB
stage6.xml	29/5/2019 22:49	Documento XML	2 KB
stage7.xml	29/5/2019 22:49	Documento XML	2 KB
stage8.xml	29/5/2019 22:49	Documento XML	2 KB
stage9.xml	29/5/2019 22:50	Documento XML	2 KB
stage10.xml	29/5/2019 22:51	Documento XML	2 KB
stage11.xml	29/5/2019 22:53	Documento XML	2 KB
stage12.xml	29/5/2019 22:56	Documento XML	3 KB
stage13.xml	29/5/2019 23:05	Documento XML	3 KB
stage14.xml	29/5/2019 23:22	Documento XML	2 KB
stage15.xml	30/5/2019 0:01	Documento XML	2 KB
stage16.xml	30/5/2019 1:21	Documento XML	2 KB
stage17.xml	30/5/2019 4:08	Documento XML	2 KB

**Figura 25-2:** Archivo XML generado al final del Cascade  
Realizado por: Ludeña, J. 2019

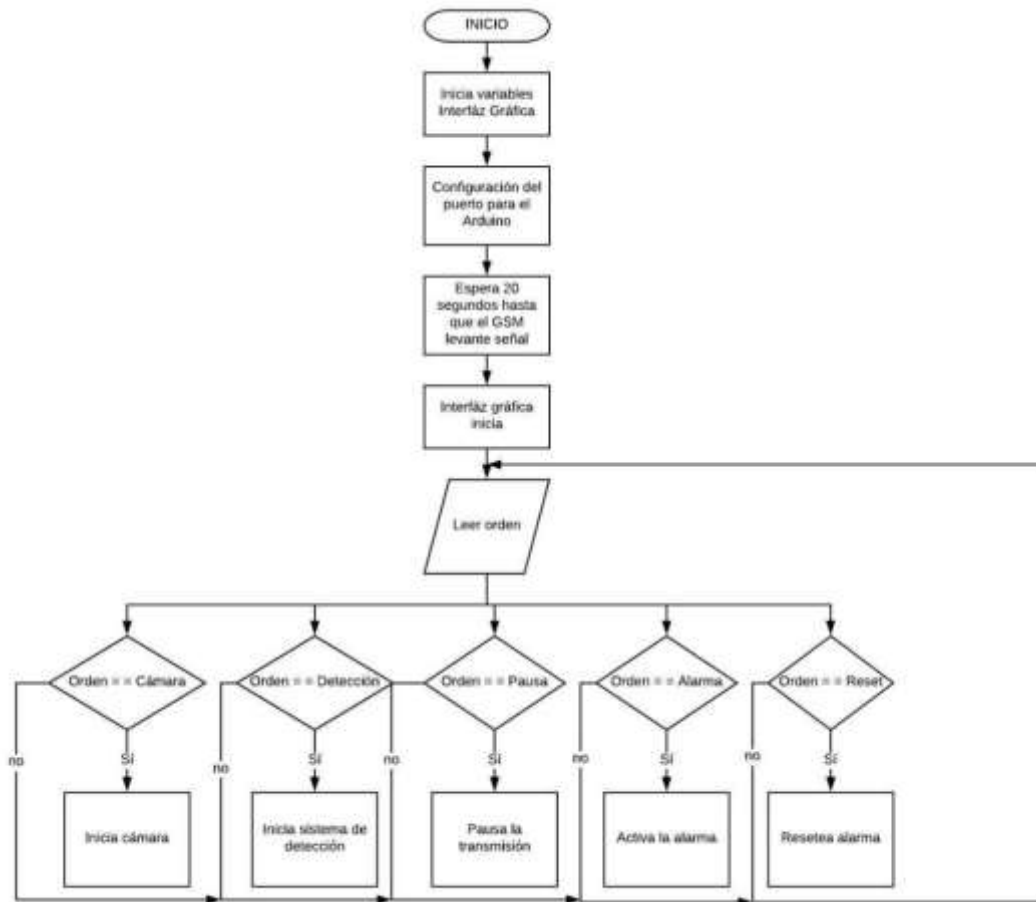
### 2.5.2 Comprobación del entrenamiento

Para comprobar si el Cascade tuvo éxito en el entrenamiento en la guide Cascade Trainer GUI tiene su propio comprobador de Cascade el cual permite subir videos o imágenes configurando el tamaño de las muestras y los vecinos más cercanos. La salida es un video o conjunto de imágenes.



**Figura 26-1:** Comprobación del entrenamiento  
Realizado por: Ludeña, J. 2019

## 2.6 Programación Python



**Figura 27-2:** Diagrama de programación en Python

Realizado por: Ludeña, J. 2019

Al iniciar el programa Python se ejecutan las librerías de Open CV, Pyqt5 entre otras que sirven para la ejecución correcta del programa. Además, se inician principalmente 2 variables fundamentales para la aplicación como son: La interfaz gráfica y la comunicación de la placa de control, la alarma controlado por un Arduino UNO las características se visualizan en la tabla 4-1.

Para el arranque de la interfaz gráfica se inicia o declaran los botones y los elementos de visualización. La placa se inicia después de configurar la interfaz, se debe señalar que para que se abra la interfaz debe pasar un tiempo aproximado de 20 segundos cuyo tiempo es el necesario para que la placa levante señal del GSM para las notificaciones y se visualice una ventana emergente contando el tiempo restante.

A continuación, se abre la pantalla principal donde se encuentran los botones con diferentes acciones como se visualiza en el diagrama 16-2 y en la figura 13-2. Cada botón tiene una función independiente como por ejemplo el botón cámara habilita la visualización de imagen que proyecta

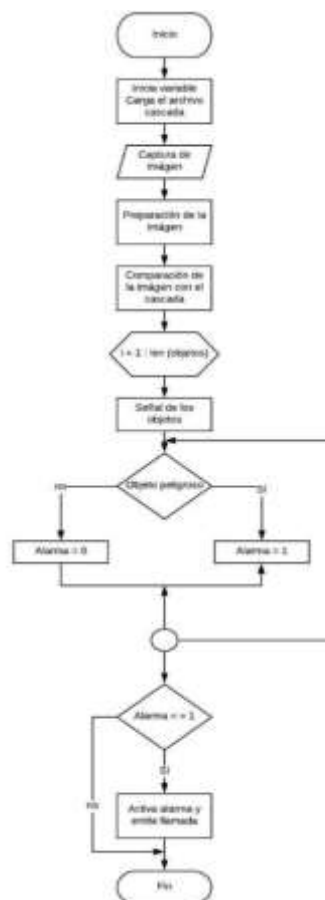
el label principal de la pantalla. La función principal del algoritmo es activar los botones de detección y alerta.

## 2.7 Sistema detección y alarma

Se emplea el sistema de detección de objetos por el método del Haar Cascade, para realizar el sistema de detección y alarma se deben seguir varios pasos y emplear el Cascade Training GUI para la realización del mismo. (Ver figura 27-2)

Se declaran variables y se carga el archivo Haar Cascade a continuación se lee la imagen de la cámara y es tratada con un filtro gaussiano transformándola a escala de grises para el ingreso al Cascade.

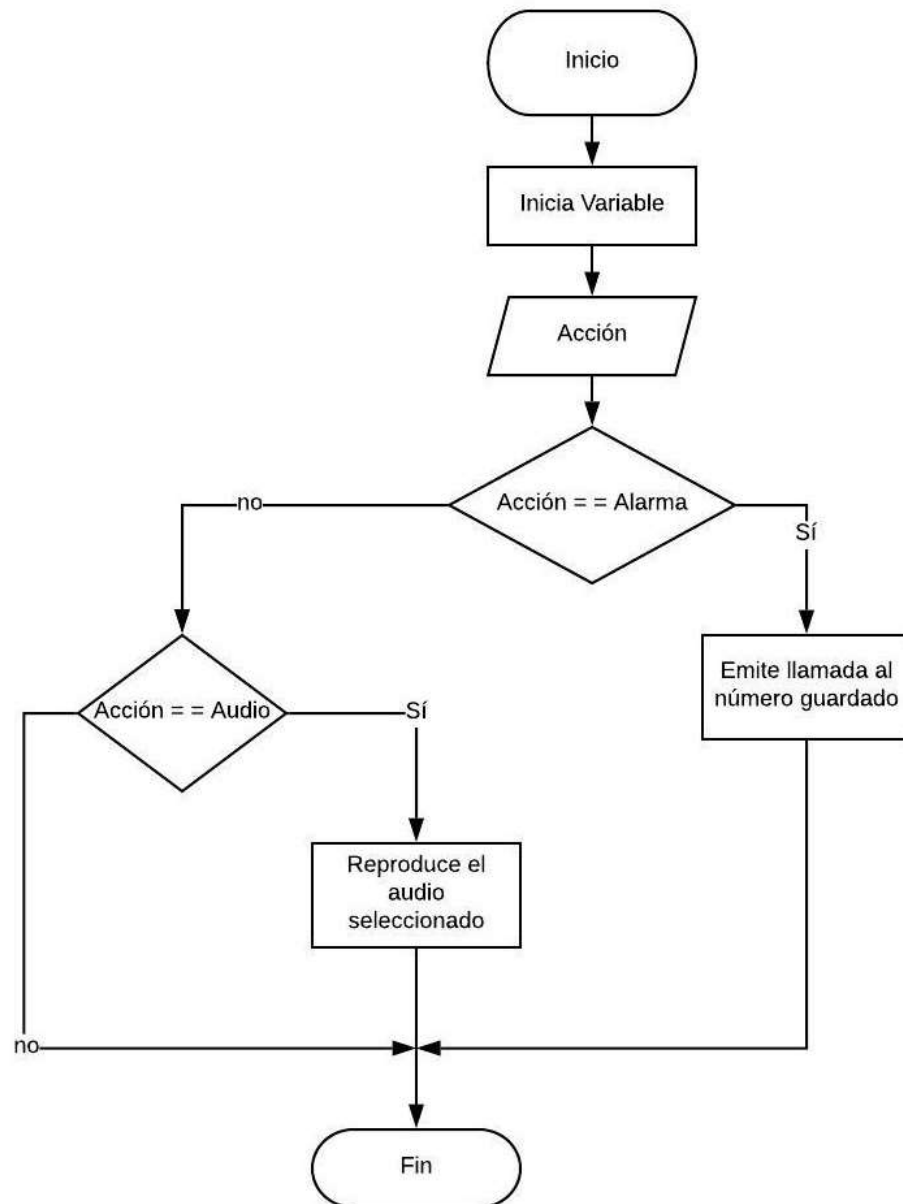
El Cascade como resultado da un vector con coordenadas X, Y, y el ancho de la imagen donde se encuentra un objeto u objetos después de esto los datos son llevados a un For donde encierra con un cuadrado al elemento y se escribe la palabra objeto dentro de la pantalla además las coordenadas de los objetos son comparadas con puntos de referencia para la alarma, si las coordenadas son mayores que la referencia entonces se enciende la alarma y así activa todos los sistemas anteriormente mencionados.



**Figura 28-2:** Algoritmo del sistema de detección y alarma  
Realizado por: Ludeña, J. 2019



## 2.8 Programación en Arduino



**Figura 29-2:** Diagrama de funcionamiento del algoritmo

Realizado por: Ludeña, J. 2019

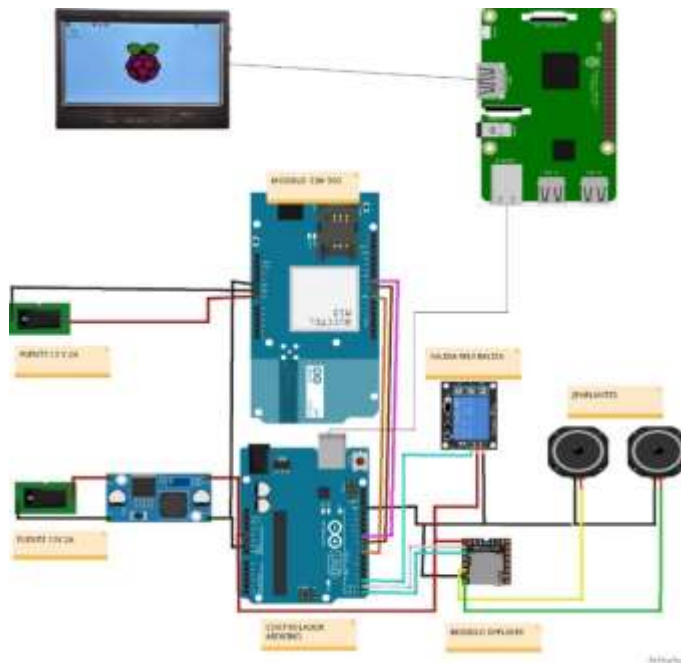
Para la programación en Arduino la comunicación que realiza este dispositivo y la Raspberry Pi es por medio de comunicación serial para ellos se envía caracteres especiales y específicos para cada acción que quiere realizar los cuales se describen en la tabla 8-2.

**Tabla 8-2:** Comandos empleados en la programación del Arduino UNO

CARÁCTER	ACCIÓN (Comandos de voz)
A	Inicio
B	Inicia cámara
C	Alarma activada
D	Alarma desactivada
E	Sistema activado
F	Alarma desactivada
G	Pausa
H	Activación de alarma
Z	Activa llamada

**Realizado por:** Ludeña, J. 2019

## 2.7 Diagrama de control



**Figura 30-2:** Esquema de diagrama de control

**Elaborado por:** Ludeña, J. 2019

Como se observa en la figura 29-2 el esquema de diagrama de control está compuesto por la Raspberry Pi, Arduino UNO, módulo SIM 900 y el DF Player. La conexión principal del sistema se realiza entre la Raspberry y el Arduino mediante comunicación serial, además la mayoría de componentes que se comunican al Arduino emplean la misma comunicación con diferentes conexiones de pines.

## CAPÍTULO III

### 3. Marco de Resultados

Concluido el diseño del sistema de seguridad para niños de 2 a 4 años se procede a realizar las pruebas de funcionamiento de la programación. Se realizan comparaciones entre la respuesta del entrenamiento del sistema por medio de la Raspberry Pi y por medio del computador en espacios controlados y no controlados.

#### 3.1 Pruebas del clasificador

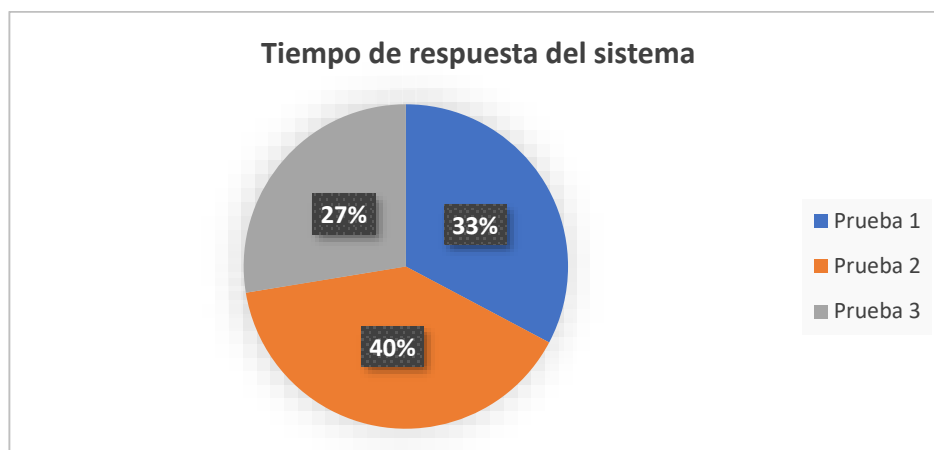
##### 3.1.1 Mediante Raspberry Pi en espacios controlados

Se efectuaron tres pruebas del clasificador en espacios controlados de los parámetros importantes que realiza el sistema como tiempo de respuesta del sistema, número de objetos reconocidos, distancia de objetos y tiempo de respuesta de llamada.

**Tabla 1-3:** Prueba del sistema mediante Raspberry Pi en espacios controlados

PARÁMETRO	TIEMPO		
	Prueba 1	Prueba 2	Prueba 3
Tiempo de respuesta del sistema	1,9s	2,3s	1,6s
Número de objetos reconocidos	2	7	8
Distancia de los objetos	1,7m	1,3m	0,9m
Tiempo de respuesta de llamada	4,9s	4,7s	4,0s

Realizado por: Ludeña, J. 2019



**Gráfico 1-3:** Tiempo de respuesta mediante Raspberry Pi en espacios controlados

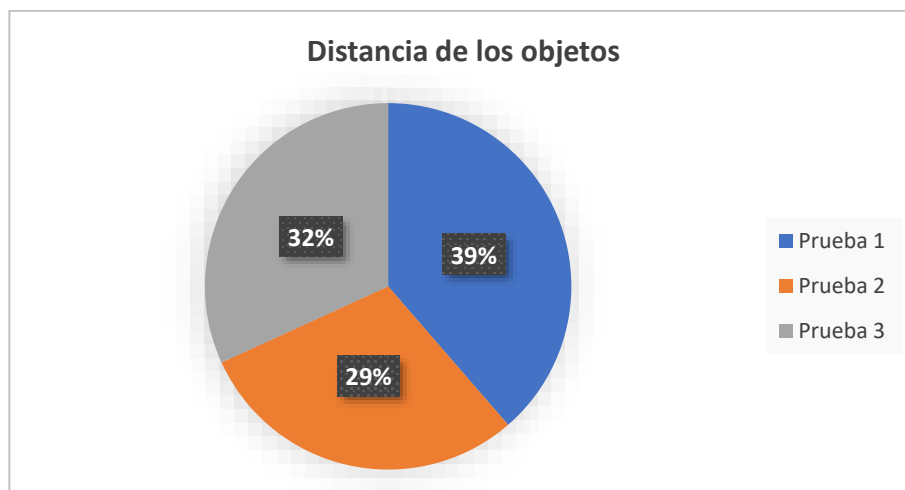
Realizado por: Ludeña, J. 2019

Al realizar las pruebas del tiempo de respuesta del sistema mediante la Raspberry Pi en espacios controlados se determina que el tiempo mínimo de espera es de 1,6 segundos y el tiempo máximo es de 2,3 segundos.



**Gráfico 2-3:** Número de objetos mediante Raspberry Pi en espacios controlados  
Realizado por: Ludeña, J. 2019

Los números de objetos detectados mediante la Raspberry Pi en espacios controlados es de 7 elementos reconocidos por el sistema.



**Gráfico 3-3:** Distancia de objetos mediante Raspberry Pi en espacios controlados  
Realizado por: Ludeña, J. 2019

La detección de objetos se ve afectado directamente por la distancia en la que se encuentra, es decir mientras más lejos está ubicado el elemento de la cámara menor será la capacidad de detección. Mediante esta prueba se determina que, en una distancia de 1,7m los objetos detectados son 2.



**Gráfico 4-3:** Tiempo de respuesta de llamada en Raspberry Pi en espacios controlados  
Realizado por: Ludeña, J. 2019

Al emitir llamadas mediante Raspberry Pi es regular en todas las pruebas, es decir no sufre mayor variación en el tiempo de llamada.

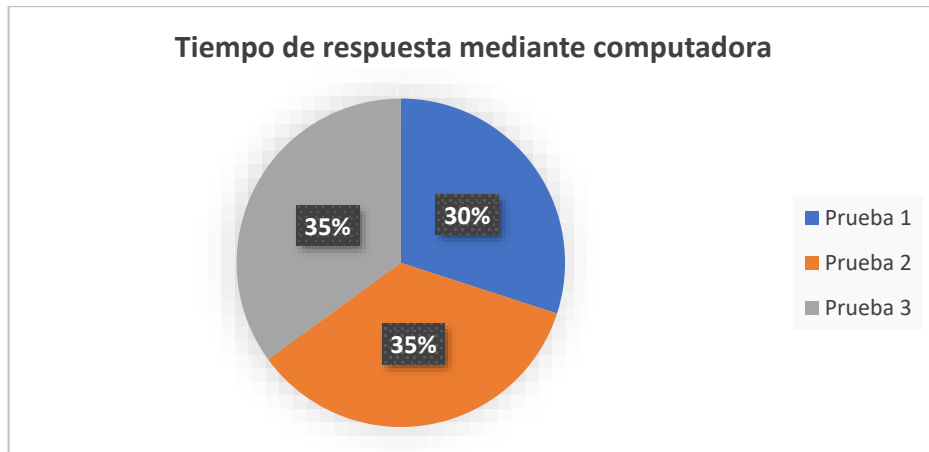
### 3.1.2 Mediante computadora en espacios controlados

Se realizaron tres pruebas del sistema mediante computadora en espacios controlados, estos datos permiten establecer una comparación entre el desarrollo del sistema mediante la Raspberry Pi y la computadora.

**Tabla 2-3:** Prueba de funcionamiento mediante computadora en espacios controlados

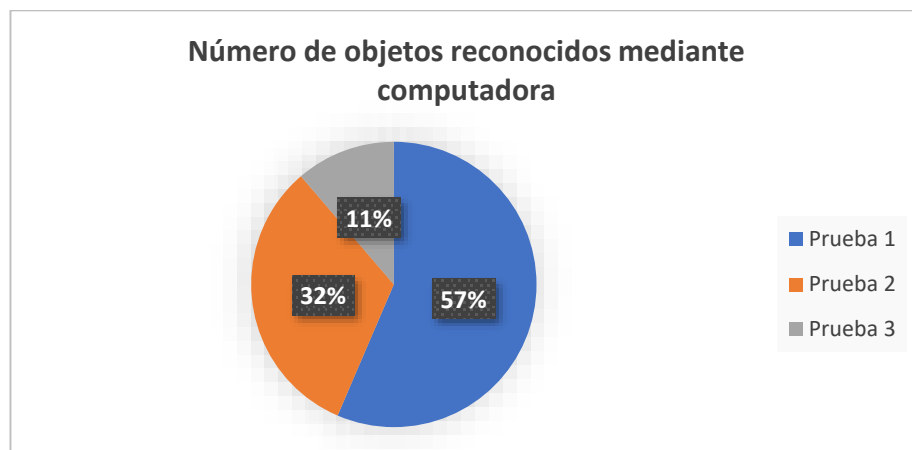
PARÁMETRO	TIEMPO		
	Prueba 1	Prueba 2	Prueba 3
Tiempo de respuesta	1,2s	1,4s	1,8s
Número de objetos reconocidos	7	4	8
Distancia de los objetos	1,5m	1,0m	0,5m
Tiempo de respuesta de llamada	4,6s	4,4s	3,8s

Realizado por: Ludeña, J. 2019



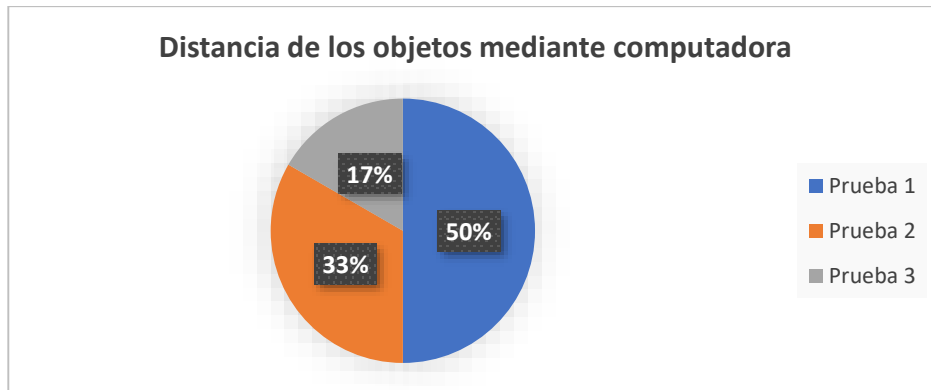
**Gráfico 5-3:** Tiempo de respuesta por computador en espacios controlados  
Realizado por: Ludeña, J. 2019

Se puede notar que el tiempo de respuesta del sistema mediante computador en espacios controlados disminuye varios segundos en comparación a las pruebas realizadas con la Raspberry Pi.



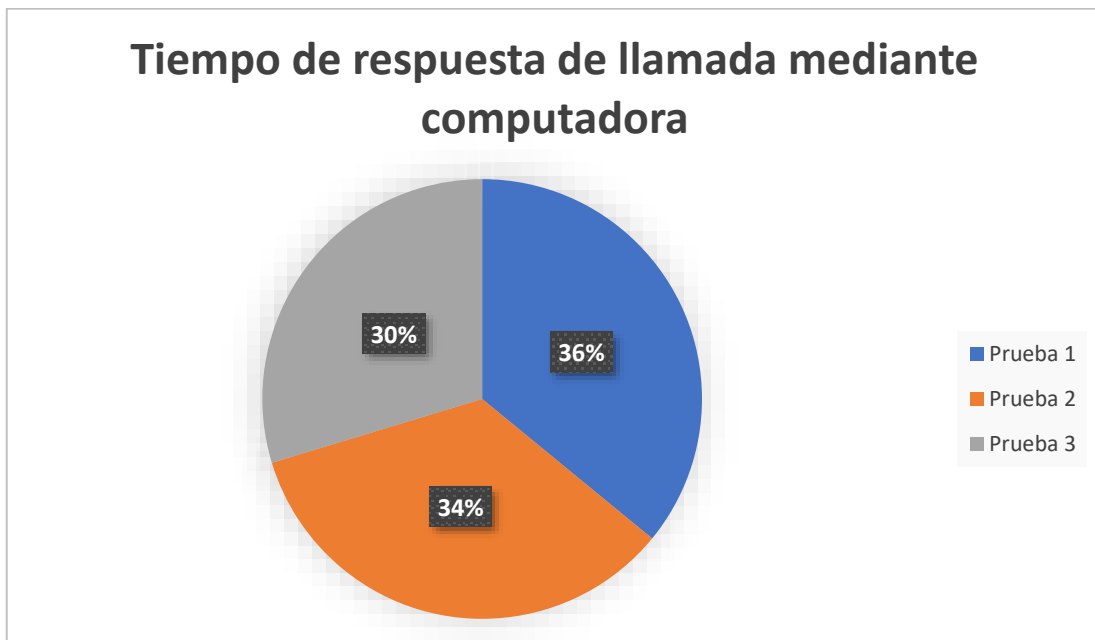
**Gráfico 6-3:** Número de objetos por computador en espacios controlados  
Realizado por: Ludeña, J. 2019

El número de objetos reconocidos mediante computadora en espacios controlados alcanza hasta 8 elementos.



**Gráfico 7-3:** Distancia de objetos mediante computadora en espacios controlados  
**Realizado por:** Ludeña, J. 2019

Mientras más corta sea la distancia mayor es el número de objetos reconocidos, en este caso la distancia fue de 0,5 y los objetos reconocidos fueron 8.



**Gráfico 8-3:** Tiempo de respuesta de llamada por computador en espacios controlados  
**Realizado por:** Ludeña, J. 2019

El tiempo de respuesta de llamada es poco variante, es decir, mantiene los rangos de tiempos. En este caso el tiempo más corto es de 3,8 segundos.

### ***3.1.3 Mediante Raspberry Pi en espacios no controlados***

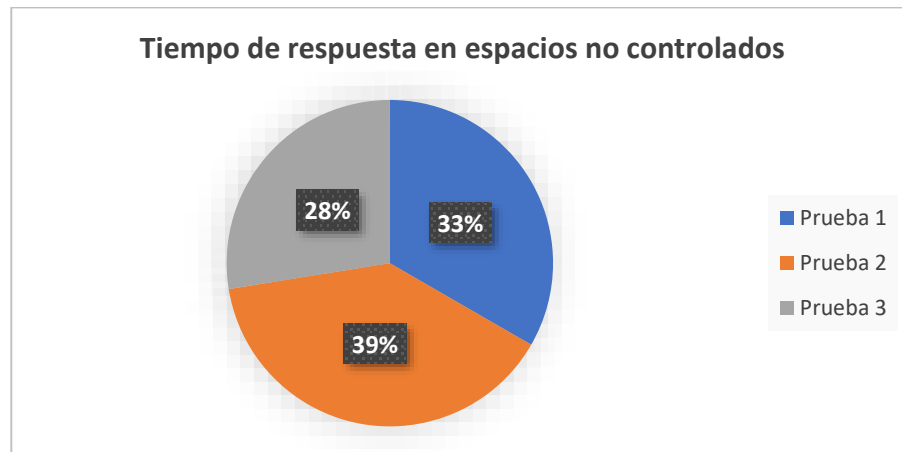
Es necesario realizar pruebas de funcionamiento en espacios no controlados para obtener datos de respuesta del sistema mediante la Raspberry Pi en espacios no controlados.



**Tabla 3-3:** Prueba de funcionamiento mediante Raspberry Pi en espacios no controlados

PARÁMETRO	TIEMPO		
	Prueba 1	Prueba 2	Prueba 3
Tiempo de respuesta	2,3 s	2,7s	1,9s
Número de objetos reconocidos	2	3	3
Distancia de los objetos	1,5	0,5m	1,2m
Tiempo de respuesta de llamada	4,4s	4,3s	4,8s

Realizado por: Ludeña, J. 2019



**Gráfico 9-3:** Tiempo de respuesta mediante Raspberry Pi en espacios no controlados

Realizado por: Ludeña, J. 2019

Los espacios no controlados hacen que el tiempo de respuesta del sistema en Raspberry sea mayor en comparación a los espacios controlados. Es decir, en esta prueba el resultado fue de 1,9 segundos.



**Gráfico 10-3:** Número de objetos por Raspberry Pi en espacios no controlados

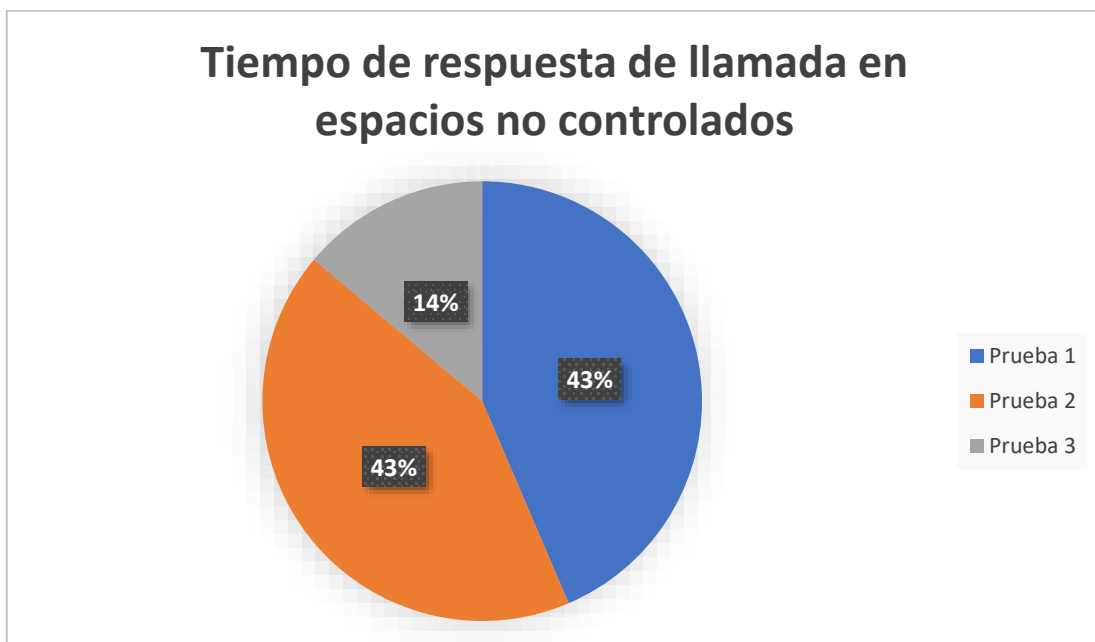
Realizado por: Ludeña, J. 2019

Los objetos reconocidos disminuyen de manera notable en comparación a los elementos reconocidos en espacios controlados, como se puede notar en el gráfico 10-3 el mayor número de objetos es tres.



**Gráfico 11-3:** Distancia de objetos mediante Raspberry Pi en espacios no controlados  
Realizado por: Ludeña, J. 2019

Como se nota en las pruebas debido al espacio no controlado la distancia de los objetos hace que sea difícil de ubicarlos, pese a esto mediante la Raspberry se encontraron 3 objetos peligrosos.



**Gráfico 12-2:** Tiempo de respuesta por Raspberry Pi en espacios no controlados  
Realizado por: Ludeña, J. 2019

En esta prueba el mayor tiempo de espera es de 4,8, pero como se ha visto en las pruebas anteriores estos valores pueden variar con intervalos de rangos entre los 0,3 segundos.

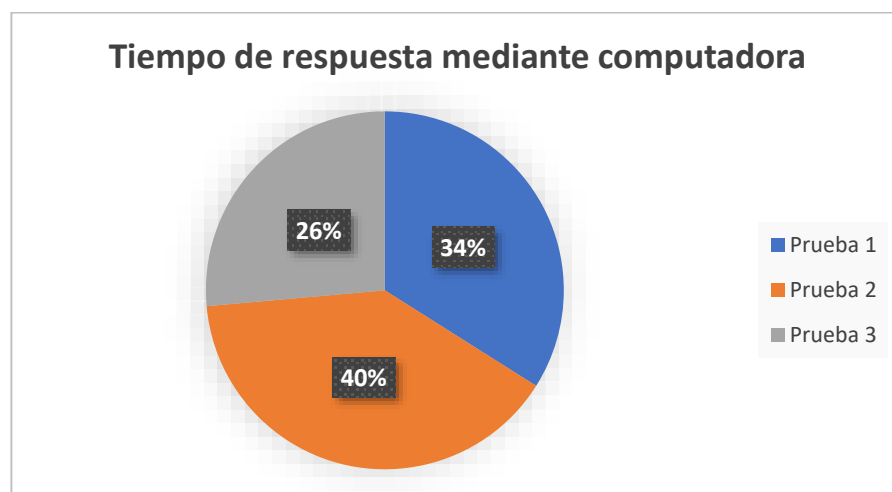
### 3.1.4 Mediante computadora en espacios no controlados

La prueba del funcionamiento del sistema mediante computadoras permite conocer la factibilidad de ejecución en espacios no controlados.

**Tabla 4-3:** Prueba del sistema mediante computadora en espacios no controlados

PARÁMETRO	TIEMPO		
	Prueba 1	Prueba 2	Prueba 3
Tiempo de respuesta	1,8 s	2,1s	2,0s
Número de objetos reconocidos	3	5	7
Distancia de los objetos	1,6	1,5m	1,0m
Tiempo de respuesta de llamada	4,3s	4,6s	4,5s

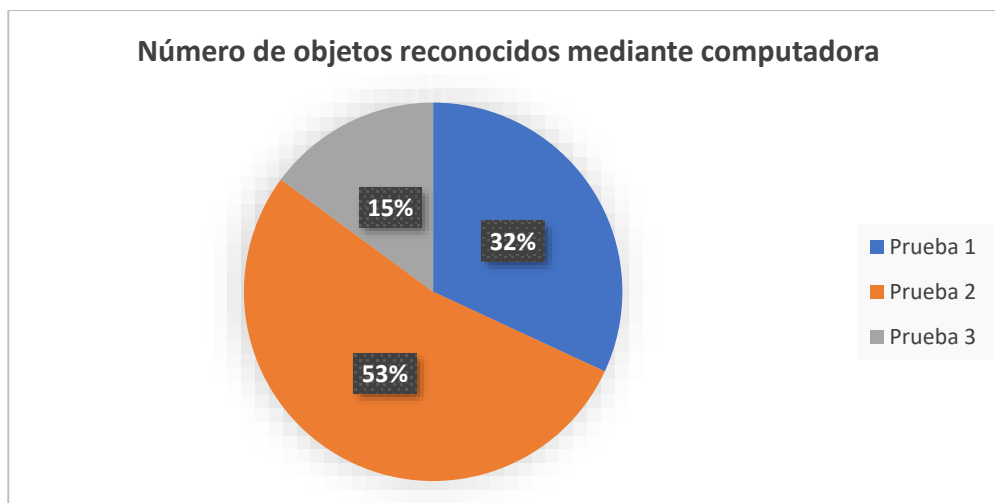
Realizado por: Ludeña, J. 2019



**Gráfico 13-3:** Tiempo de respuesta computadora en espacios no controlados

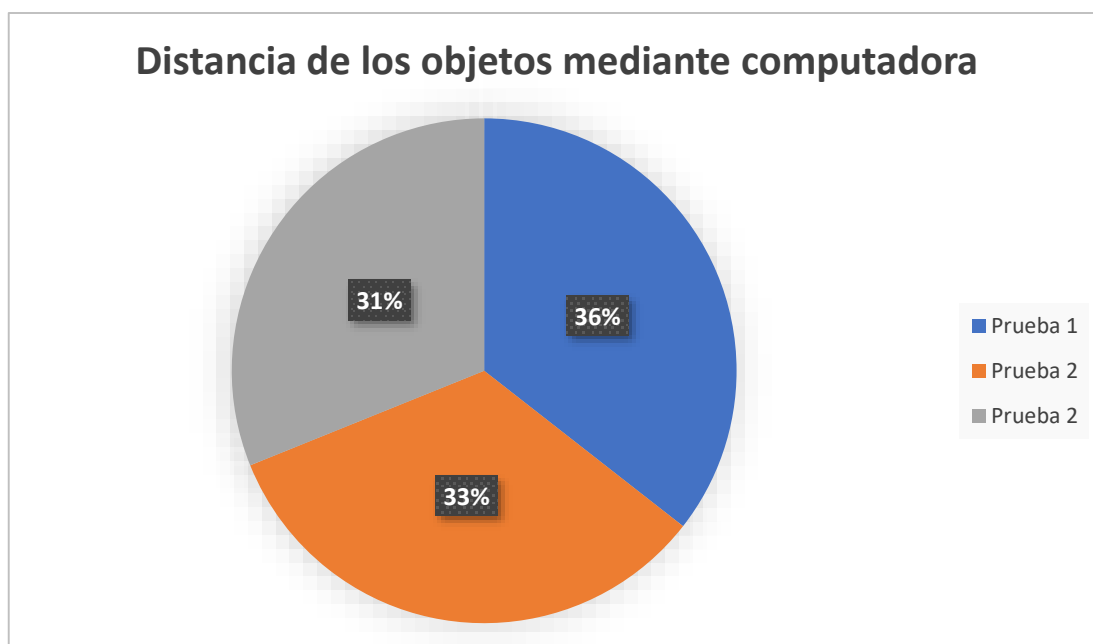
Realizado por: Ludeña, J. 2019

En esta prueba se puede notar que la respuesta del sistema mediante computadora en espacios no controlados es de 1,8 segundos la mínima y la máxima de 2,1 segundos.



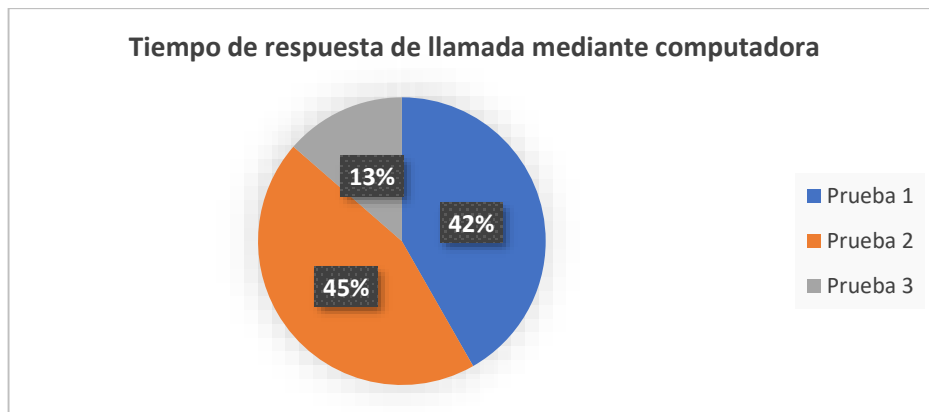
**Gráfico 14-3:** Número de objetos por computadora en espacios no controlados  
**Realizado por:** Ludeña, J. 2019

Los objetos reconocidos por el sistema mediante computadora manejan un intervalo de 3, 5 y 7 elementos esto depende de la distancia en la que están ubicados.



**Gráfico 15-3:** Distancia de objetos mediante computadora en espacios no controlados  
**Realizado por:** Ludeña, J. 2019

La distancia de los objetos debe ser más corta para que el sistema detecte los elementos, en este caso la distancia establecida es de 1m y se detectaron 7 objetos.



**Gráfico 16-3:** Tiempo de respuesta mediante computadora en espacios no controlados  
Realizado por: Ludeña, J. 2019

El tiempo de respuesta no sufre mayor variación, en esta prueba el tiempo mínimo fue de 4,3 y el máximo de 4,6.

### 3.2 Comparación de resultados entre Raspberry y computador

Con las pruebas anteriores se obtuvieron los datos que permiten las comparaciones entre los sistemas mediante Raspberry Pi y computadora en espacios controlados y viceversa.

#### 3.2.1 Espacios controlados

Con los datos obtenidos anteriormente se puede establecer una clara diferencia en el funcionamiento del sistema entre la Raspberry Pi y el computador, el sistema de seguridad funciona de mejor manera cuando es ejecutado desde la computadora. A continuación, se describe en la tabla 5-3. (Ver anexo 6)

**Tabla 5-3:** Resultado de pruebas de funcionamiento en espacios controlados

PARÁMETROS	COMPUTADOR	RASPBERRY PI
Tiempo de respuesta	1,46	1,93
Número de objetos reconocidos	6	6
Distancia de los objetos	1	1,3
Tiempo de respuesta de llamada	4,26	4,53

Realizado por: Ludeña, J. 2019

### 3.2.2 Espacios no controlados

Como se mencionó anteriormente las pruebas del sistema en espacios no controlados determinan la factibilidad de aplicar el sistema en lugares indeterminados, el análisis de los distintos parámetros permite la comparación de respuesta entre el computador y la Raspberry Pi demostrando que mediante computadora se desarrolla de mejor manera el sistema de seguridad, como se puede ver en el Anexo 7

**Tabla 6-3:** Resultado de pruebas de funcionamiento en espacios no controlados

PARÁMETROS	COMPUTADOR	RASPBERRY PI
Tiempo de respuesta	1,96	2,3
Número de objetos reconocidos	5	3
Distancia de los objetos	1,36	1,06
Tiempo de respuesta de llamada	4,46	4,5

Realizado por: Ludeña, J. 2019

## 3.3 Prueba de detección de objetos a distancia

### 3.3.1 Detección de objetos en espacios controlados

Como se explicó anteriormente, para la detección de objetos es necesario mantener una distancia mínima de 0,5m y máxima de 1,5m, sin embargo, en una distancia de 1,7m hasta 1,9m los objetos detectados irán disminuyendo mientras que en una distancia de 2m será casi nula la capacidad de reconocimiento de los objetos.

**Tabla 7-3:** Detección de objetos en espacios controlados

OBJETOS	DISTANCIA / DETECCIÓN								
	0,5m	0,9m	1,0m	1,2m	1,5m	1,7m	1,8m	1,9m	2,0m
Tijera	Si	Si	Si	Si	Si	Si	Si	Si	No
Estilete	Si	Si	Si	Si	Si	Si	Si	Si	No
Botella	Si	Si	Si	Si	Si	Si	Si	Si	No
Cuchillo	Si	Si	Si	Si	Si	Si	Si	No	No

Tomacorriente	Si	Si	Si	Si	Si	Si	No	No	No
Latas	Si	Si	Si	Si	Si	No	No	No	No
Pinzas	Si	Si	Si	Si	Si	No	No	No	No
Marcadores	Si	Si	Si	Si	Si	Si	Si	No	No

Realizado por: Ludeña, J. 2019

### 3.3.2 Detección de objetos en espacios no controlados

Para la detección de objetos en espacios no controlados la distancia optima esta desde los 0,5m hasta los 1,2m mientras que si maneja distancias de 1,5 hasta 1,8 el número de objetos reconocidos disminuye, en distancias de 1,9m y 2,0m el sistema no detecta ningún objeto.

**Tabla 8-3:** Detección de objetos en espacios no controlados

OBJETOS	DISTANCIA / DETECCIÓN								
	0,5m	0,9m	1,0m	1,2m	1,5m	1,7m	1,8m	1,9m	2,0m
Tijera	Si	Si	Si	Si	Si	Si	No	No	No
Estilete	Si	Si	Si	Si	Si	No	No	No	No
Botella	Si	Si	Si	Si	Si	Si	Si	No	No
Cuchillo	Si	Si	Si	Si	Si	Si	No	No	No
Tomacorriente	Si	Si	Si	Si	No	No	No	No	No
Latas	Si	Si	Si	Si	Si	No	No	No	No
Pinzas	Si	Si	Si	Si	No	No	No	No	No
Marcadores	Si	Si	Si	Si	No	No	No	No	No

Realizado por: Ludeña, J. 2019

### 3.4 Censo de carga

Se tomaron los valores de voltios y amperios de los elementos que componen la placa para determinar la potencia total que emplea el sistema de seguridad, siendo esto un total de 39,14W utilizado para el correcto funcionamiento del sistema permitiendo así cumplir todas las funciones establecidas para cada componente.

**Tabla 9-3:** Censo de potencia

Componente	Voltio	Amperio	Potencia
Arduino	5,01	2,0	10,02
Raspberry Pi	5,03	1,94	9,75
SIM 900	12	0,8	9,6
DF Player	5,02	1	5,02
Relé, parlantes y demás componentes	5,01	0,95	4,75
	<b>TOTAL</b>	<b>6,69 A</b>	<b>39,14 W</b>

**Realizado por:** Ludeña, J. 2019

En caso de existir fallos de energía y de acuerdo a la potencia obtenida en la tabla 9-3 se puede colocar un sistema de back up de energía existente en el mercado, el cual consiste en un inversor mínimo de 200W y la batería de 12V. Si la corriente del sistema es de 1.12A medidos AC la corriente de la batería dependerá del tiempo de trabajo del sistema que necesita el usuario.

Mediante la fórmula:

$$h = \frac{\text{corriente batería}}{\text{corriente del sistema}}$$

**Tabla 10-3:** Tiempo de duración de baterías estándar

CORRIENTE	TIEMPO DE DURACIÓN
12V – 18ah	2,690
12V – 12ah	1,793
12V – 10ah	1,494
12V – 7ah	1,046
6V – 4ah	0,597

**Realizado por:** Ludeña, J. 2019



### 3.5 Presupuesto

Para el desarrollo del sistema de seguridad para niños de 2 a 4 años usando visión artificial se invierte alrededor de \$508,00 dólares, en comparación a otros sistemas que existen actualmente es accesible tomando en cuenta que los sistemas antes mencionados no cumplen la función de alertar si el niño se encuentra en peligro.

**Tabla 11-3:** Costos

COMPONENTE	CANTIDAD	VALOR (\$)
Raspberry Pi 3	1	\$ 108,00
Arduino UNO	1	\$ 11,00
Módulo GSM SIM 900	1	\$ 35,00
Módulo DF Player Mini	1	\$ 8,00
Módulo DC-DC Setp Down Kim 0551	1	\$ 4,50
Relé de un canal	1	\$ 3,50
Baliza	1	\$ 12,00
Pantalla	1	\$ 145,00
Cámara Logitech C270	1	\$ 43,00
Parlantes Anera	1	\$ 8,00
Fuente 12 V	1	\$ 12,00
Fuente 5 V	1	\$ 8,00
Cables de conexión (entre otros)	1	\$ 50,00
Corte laser	1	\$ 60,00
	<b>Total</b>	<b>\$ 508,00</b>

**Realizado por:** Ludeña, J. 2019

## CONCLUSIONES

- El funcionamiento del sistema de seguridad para niños de 2 a 4 años permite reducir un 20% de accidentes que existe en la actualidad, mediante la visión artificial y los algoritmos de detección se crea sistemas que ayudan con la supervisión y seguridad de los niños en todos los entornos que se encuentren.
- Para que la implementación del sistema es indispensable identificar los hardware y software necesarios, en este caso fueron: Raspberry Pi, Arduino UNO, DF Player, Módulo SIM 900, Relé entre otros, de igual forma para el desarrollo de algoritmos de detección y supervisión como el Haar Cascade, y para la creación de la interfaz gráfica el QT Designer.
- Al seleccionar los elementos para la implementación del sistema se busca reducir costos, pero sin perder la calidad es decir se seleccionan los más óptimos, para la creación del algoritmo de detección se emplea el Haar Cascade porque ocupa menos recursos al momento de iniciar el entrenamiento.
- Para la verificación del funcionamiento del sistema se realizaron 10 pruebas del prototipo y de los algoritmos de detección de los diferentes parámetros tomando en cuenta las tres pruebas con datos más sobresalientes. Después de esto, se establece que el desarrollo del sistema de seguridad es 15% más factible que los sistemas convencionales al implementarlo en los hogares para el cuidado de los niños de temprana edad, evitando que ocurran accidentes.
- Cuando el sistema detecta un objeto peligroso inmediatamente envía una señal que activa la alarma en el prototipo, conjuntamente con esto el sistema emite la llamada al número registrado en la base de datos. Con los resultados obtenidos se evalúa que el sistema es efectivo en espacios controlados y no posee una gran respuesta en espacios no controlados.

## RECOMENDACIONES

- Realizar una investigación profunda de todos los sistemas de seguridad que existen en la actualidad, para recoger información que ayude de punto de partida la creación de nuevos sistemas de supervisión aplicando visual artificial.
- Realizar una retroalimentación de todas las fases, componentes y algoritmos que se van a implementar para que no muestre fallas en algún punto del desarrollo o en la fase final de la creación del sistema de seguridad.
- Determinar y establecer todos los componentes a utilizar en el desarrollo del prototipo, para el desarrollo del software determinar el conjunto de imágenes positivas y negativas que se vayan almacenando en una base de datos para facilitar y reducir el tiempo de entrenamiento.
- Realizar varias pruebas para evaluar el funcionamiento del sistema, para que el tiempo de entrenamiento sea mínimo se debe emplear una computadora que posea características mínimas como memoria RAM de 16Gb y procesador Core i7.
- Para crear sistemas de supervisión se puede emplear distintos softwares como Matlab, Java (Android Studio) de igual manera desarrollar aplicaciones móviles que cumplan la misma función, esto depende del entorno en que se desea aplicarlos sin alterar el porcentaje de eficacia.

## GLOSARIO

<b>Arrays:</b>	Guarda un conjunto de objetos de la misma clase.
<b>Bit:</b>	Unidad de medida de información.
<b>Bucle:</b>	Secuencia que ejecuta repetidas veces un código.
<b>Buffer:</b>	Espacio en un disco o en un instrumento digital reservado para almacenamiento.
<b>Coaxial:</b>	Cable que se utiliza para transmitir señales de electricidad de alta frecuencia.
<b>Datasheet:</b>	Ficha técnica, hoja técnica u hoja de datos.
<b>Electromagnética:</b>	Combinación de campos eléctricos y magnéticos oscilantes.
<b>Ethernet:</b>	Estándar de redes de área local para computadores.
<b>Fotosensible:</b>	Que es sensible a la acción de la luz.
<b>Ironpython:</b>	Implementación del intérprete Python, escrita totalmente en C#.
<b>Jpython:</b>	Lenguaje de programación de alto nivel.
<b>Monocromáticos:</b>	Colores de un solo matiz o tono.
<b>Pixel:</b>	Menor unidad homogénea en color que forma parte de una imagen digital.
<b>Semiconductores:</b>	Elementos que tienen una conductividad eléctrica inferior a la de un conductor metálico pero superior a la de un buen aislante.
<b>Umbral:</b>	Cantidad mínima de señal que ha de estar presente para ser registrada por un sistema.
<b>Vidicón:</b>	Tubo de vacío capaz de convertir imágenes luminosas en una tensión eléctrica correspondiente al brillo de las imágenes.

## BIBLIOGRAFÍA

**ARACIL, R.** *Desarrollo de un sistema cognitivo de visión para la navegación robótica* [en línea] SURF (Speed Up Robust Features). 2012 [Consulta: 02 de febrero de 2019]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/17010/memoria.pdf?sequence=1>

**BUSTAMANTE, S.** Algoritmos de procesamiento de imagen aplicados a la detección de figuras geométricas y sus propiedades espaciales. [en línea] (Trabajo de titulación). (Postgrado) Pontificia universidad católica de Valparaíso, Facultad de ingeniería, Escuela de ingeniería informática. Chile, Valparaíso. 2014 pp. 11 – 13 [Consulta: 24 de diciembre de 2018]. Disponible en: [http://opac.pucv.cl/pucv\\_txt/txt-4500/UCE4968\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-4500/UCE4968_01.pdf)

**CHAMBA, C.** Implementación de un prototipo de sistema de georreferenciación mediante la tecnología GPS/GSM para personas vulnerables. [en línea] (Trabajo de titulación). (Postgrado) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica, Telecomunicaciones y Redes. Ecuador, Chimborazo. 2017 pp 25 [Consulta: 07 de enero de 2019]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/6367/3/98T00136.pdf>

**COGNEX.** *Introducción a la visión artificial*. [en línea] Una guía para la automatización de procesos y mejoras de calidad. 2016 [Consulta: 22 de diciembre de 2018]. Disponible en: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

**DE LA CÁMARA, R.** Arduino + módulo GSM/GPRS: monitorización, automatización y gestión remota en un viñedo. [en línea] (Trabajo de titulación). (Tecnología) Universidad Oberta de Catalunya. Grado en Tecnologías de Telecomunicación. España, Barcelona. 2017 pp 44 – 45 21 [Consulta: 07 de enero de 2019]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/65345/6/radriandTFG0617memoria.pdf>

**DUARTE, M. & CHANG, L.** *Clasificación de objetos en imágenes usando SIFT*. [en línea] Descriptores de puntos claves SIFT. 2016 pp. 3 – 5 [Consulta: 08 de febrero de 2019]. Disponible en: <http://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/chang-duarte.pdf>

**GARCIA, E.** *Visión artificial*. [en línea] Conceptos teóricos. 2016 [Consulta: 19 de diciembre de 2018]. Disponible en: [https://www.exabyteinformatica.com/uoc/Informatica/Diseno\\_de\\_interaccion/Diseno\\_de\\_interaccion\\_\(Modulo\\_5\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Diseno_de_interaccion/Diseno_de_interaccion_(Modulo_5).pdf)

**GARCÍA, J & MARCILLO, C.** Sistema domótico mediante Smartphone de la iluminación en el auditorio de la carrera de Computación-Espam-Mfl [en línea] (Trabajo de titulación). (Postgrado) Escuela Superior Politécnica Agropecuaria De Manabí Manuel Félix López, Carrera Informática. Ecuador, Manabí. 2017 pp 8 – 9 [Consulta: 02 de febrero del 2019]. Disponible en: <http://repositorio.esпам.edu.ec/bitstream/42000/664/1/TC116.pdf>

**GUERRA, F.** Diseño de un sistema de control domótico y video vigilancia supervisado por un teléfono móvil.. [en línea] (Trabajo de titulación). (Postgrado) Pontificia Universidad Católica del Peru, Facultad de Ciencias e Ingeniería. Perú, Lima. 2013 pp 20 – 21 [Consulta: 05 de enero de 2019]. Disponible en: [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5375/GUERRA\\_FELIPE\\_SISTEMA\\_CONTROL\\_DOMOTICO\\_VIDEO\\_VIGILANCIA\\_TELEFONO\\_MOVIL.pdf?sequence=1](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5375/GUERRA_FELIPE_SISTEMA_CONTROL_DOMOTICO_VIDEO_VIGILANCIA_TELEFONO_MOVIL.pdf?sequence=1)

**GONZÁLEZ, C.** Aplicaciones orientadas a la domótica con Raspberry Pi. [en línea] (Trabajo de titulación). (Postgrado) Universidad de Sevilla, Departamento de Ingeniería Electrónica, Escuela Técnica Superior de Ingeniería. España, Sevilla. 2015 pp 7 – 11 [Consulta: 26 de diciembre de 2018]. Disponible en: <http://bibing.us.es/proyectos/abreproy/90330/fichero/Memoria+TFG.pdf>

**GONZÁLEZ, I.** Diseño e implementación de sistema interactivo de información de docentes, con Raspberry Pi. [en línea] (Trabajo de titulación). (Postgrado) Universidad Politécnica Salesiana Sede Guayaquil, Facultad de Ingenierías, Ingeniería Electrónica. Ecuador, Guayaquil. 2015 pp 25 – 27 [Consulta: 03 de enero de 2019]. Disponible en: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwj4en\\_9rjgAhUJnFkKHZekDC0QFjACegQIBxAC&url=https%3A%2F%2Fdspace.ups.edu.ec%2Fbitstream%2F123456789%2F10408%2F1%2FUPS-GT001458.pdf&usg=AOvVaw1hKfBSgsx3ImB73kwDAZV1](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwj4en_9rjgAhUJnFkKHZekDC0QFjACegQIBxAC&url=https%3A%2F%2Fdspace.ups.edu.ec%2Fbitstream%2F123456789%2F10408%2F1%2FUPS-GT001458.pdf&usg=AOvVaw1hKfBSgsx3ImB73kwDAZV1)

**HIDALGO, W.** Diseño e implementación de un sistema de adquisición de señales biométricas mediante mensajes SMS [en línea] (Trabajo de titulación). (Postgrado) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica, Telecomunicaciones y Redes. Ecuador, Chimborazo. 2015 pp. 20 – 22 [Consulta: 29 de diciembre de 2018]. Disponible en: <http://dspace.espace.edu.ec/bitstream/123456789/3790/1/98T00060.pdf>

**KANNAN, R; et al.** Performance Evaluation of SIFT & FLANN and HAAR Cascade Image Processing Algorithms for Object Identification in Robotic Applications [en línea] USA: Volumen 118 No. 18 2018. pp. 2605 - 2607 [Consulta: 05 de febrero de 2019]. Disponible en: <https://acadpubl.eu/jsi/2018-118-18/articles/18c/37.pdf>

**LOZANO, G & ORDUZ, J.** Diseño de un sistema de visión artificial para la revisión del nivel de llenado de bebidas embotelladas [en línea] (Trabajo de titulación). (Postgrado) Universidad Autónoma del Caribe, Facultad de ingenierías, Ingeniería Mecatrónica. Colombia, Barranquilla. 2015 pp. 33 [Consulta: 23 de diciembre de 2018]. Disponible en: <http://repositorio.uac.edu.co/bitstream/handle/11619/1371/TMCT%200013C.pdf?sequence=1&isAllowed=y>

**MALDONADO, L.** Diseño experimental de un dispositivo que por medio de una fuente de radiación infrarroja externa permita visualizar una imagen térmica de un objetivo [en línea] (Trabajo de titulación). (Postgrado) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Control y Redes Industriales. Ecuador, Chimborazo. 2018 pp. 16 – 23 [Consulta: 24 de diciembre de 2018].

**MARTIN, S.** Desarrollo de aplicación Wireless basado en sistemas de radio integrada Easyradio [en línea] (Trabajo de titulación). (Postgrado) Universidad de Jaén, Escuela Politécnica Superior de Linares, Ingeniería de Tecnologías de Telecomunicaciones. España, Jaén. 2016 pp 37 – 38 [Consulta: 07 de enero de 2019]. Disponible en: <http://tauja.ujaen.es/bitstream/10953.1/6687/1/TFGSaraRiveroMartinderuedas.pdf>

**MASIP, D.** *El lenguaje Python*. [en línea] Instalación de Python. 2017 [Consulta: 19 de diciembre de 2018]. Disponible en: [https://www.exabyteinformatica.com/uoc/Inteligencia\\_artificial/Inteligencia\\_artificial\\_avanzada/Inteligencia\\_artificial\\_avanzada\\_\(Modulo\\_2\).pdf](https://www.exabyteinformatica.com/uoc/Inteligencia_artificial/Inteligencia_artificial_avanzada/Inteligencia_artificial_avanzada_(Modulo_2).pdf)

**REINIUS, S.** *Object recognition using the OpenCV Haar cascade-classifier on the iOS platform*. [en línea] Haar classification. 2013 pp. 12 – 13 [Consulta: 06 de febrero de 2019]. Disponible en: <http://www.diva-portal.org/smash/get/diva2:601707/fulltext01.pdf>

**SIMON.** *Seguridad en el hogar: qué debemos proteger*. [en línea] Planificando la seguridad en el hogar. 2017 [Consulta: 19 de diciembre de 2018]. Disponible en: <https://bricoladores.simonelectric.com/seguridad-en-el-hogar-que-debemos-proteger>

**UPTON, E. & HALFACREE, G.** *Raspberry Pi* [en línea] Guía del usuario. 2017 [Consulta: 07 de enero de 2019]. Disponible en: [http://88.25.236.137/files/1530867960\\_2X3tMBzWxi-33.pdf](http://88.25.236.137/files/1530867960_2X3tMBzWxi-33.pdf)

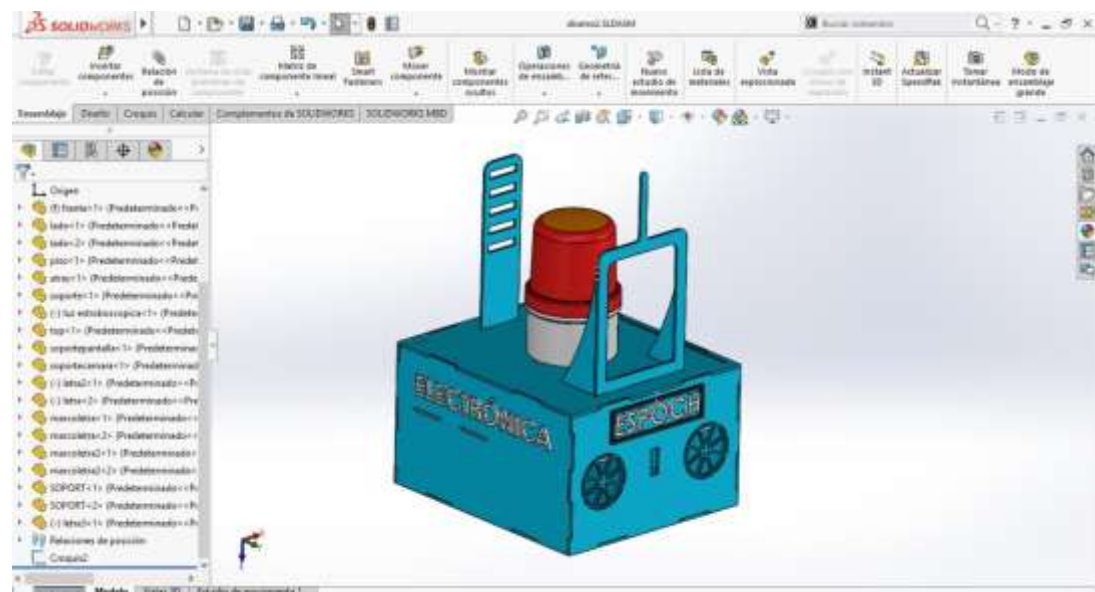
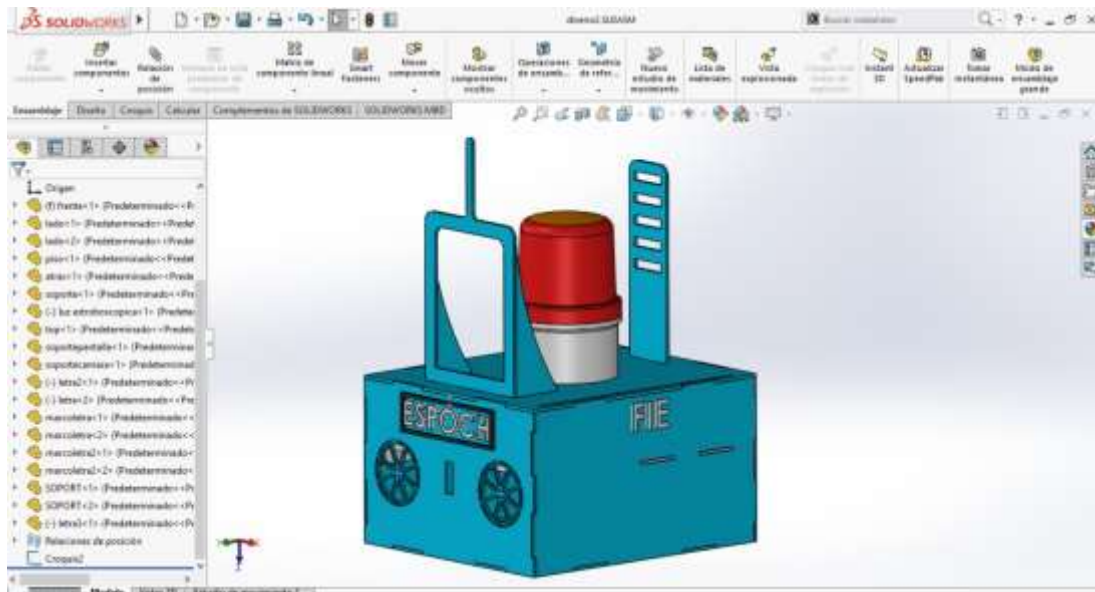
**VARGAS, V.** Sistema de Visión Artificial para el Control De Calidad en Piezas Cromadas. [en línea] (Trabajo de titulación). (Postgrado) Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica. México, México D.F. 2010 pp.11 – 13 [Consulta: 22 de diciembre de 2018]. Disponible en: <https://tesis.ipn.mx/bitstream/handle/123456789/7250/68.pdf?sequence=1&isAllowed=y>

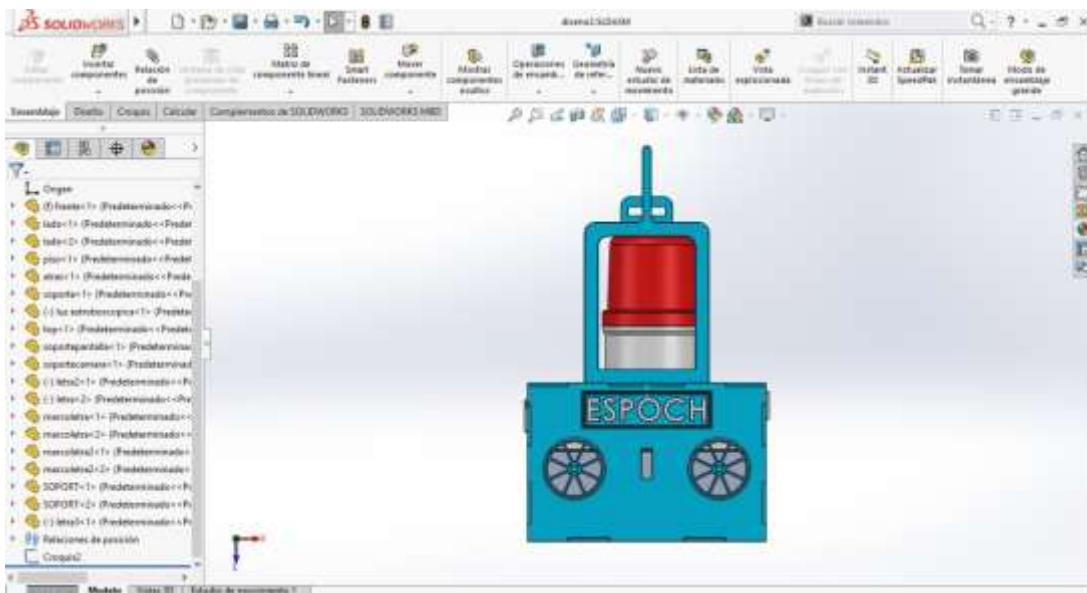
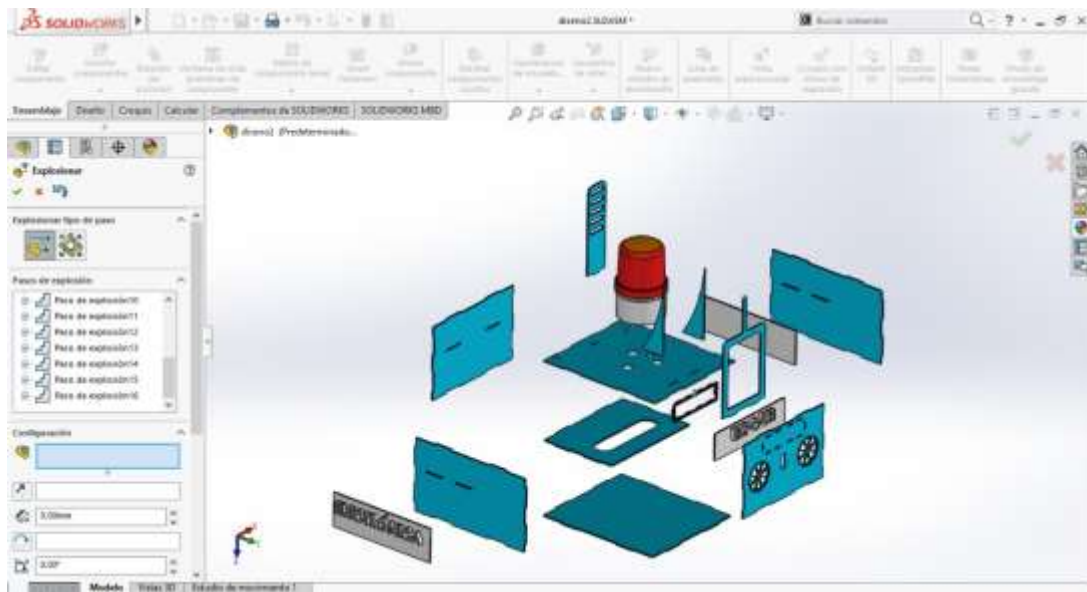
**VIERA, G.** Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao. [en línea] (Trabajo de titulación). (Postgrado) Universidad de Piura, Facultad de Ingeniería, Departamento de Ingeniería Mecánico-Eléctrica. Perú, Piura. 2017 [Consulta: 28 de diciembre de 2018]. Disponible en: [https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME\\_218.pdf?sequence=1](https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME_218.pdf?sequence=1)



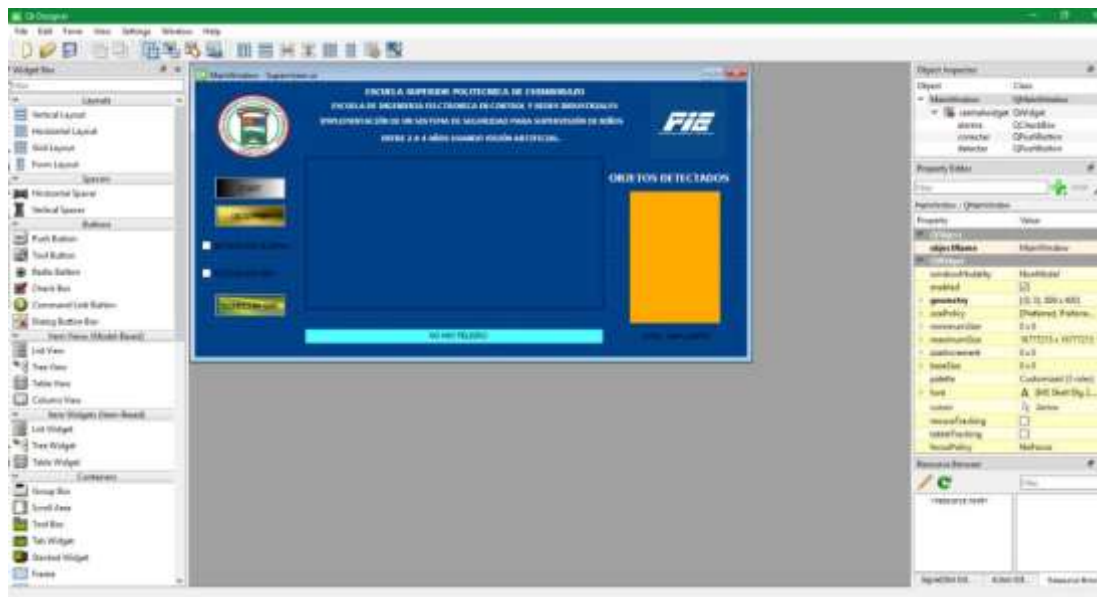
## ANEXOS

### Anexo A Creación del prototipo en Solid Works

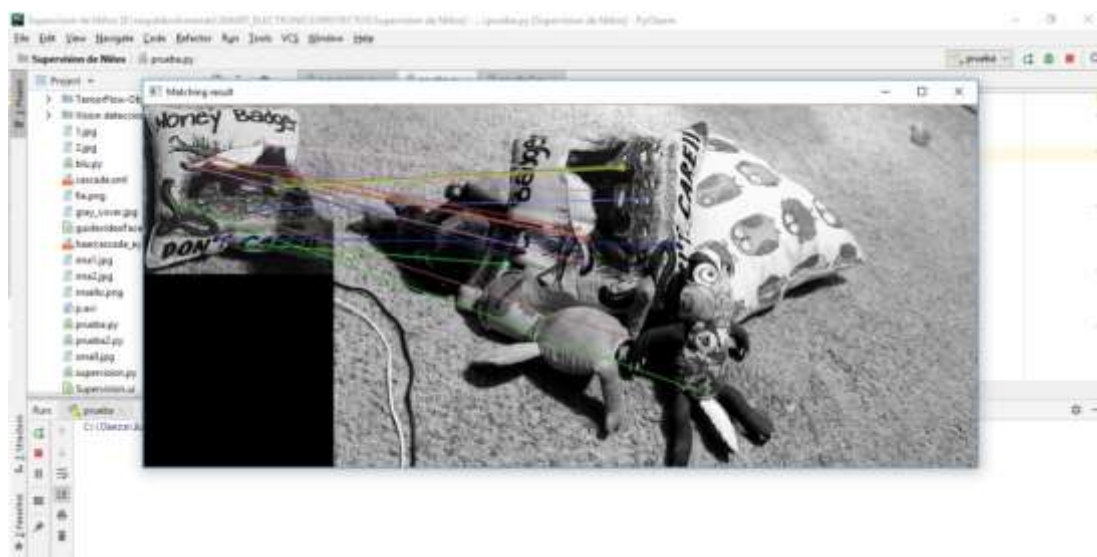




## Anexo B Pruebas de diseño de la interfaz gráfica



## Anexo C Prueba de algoritmo







## Anexo E Prueba final del sistema de supervisión





Anexo F Prueba de detección de objetos en espacios controlados





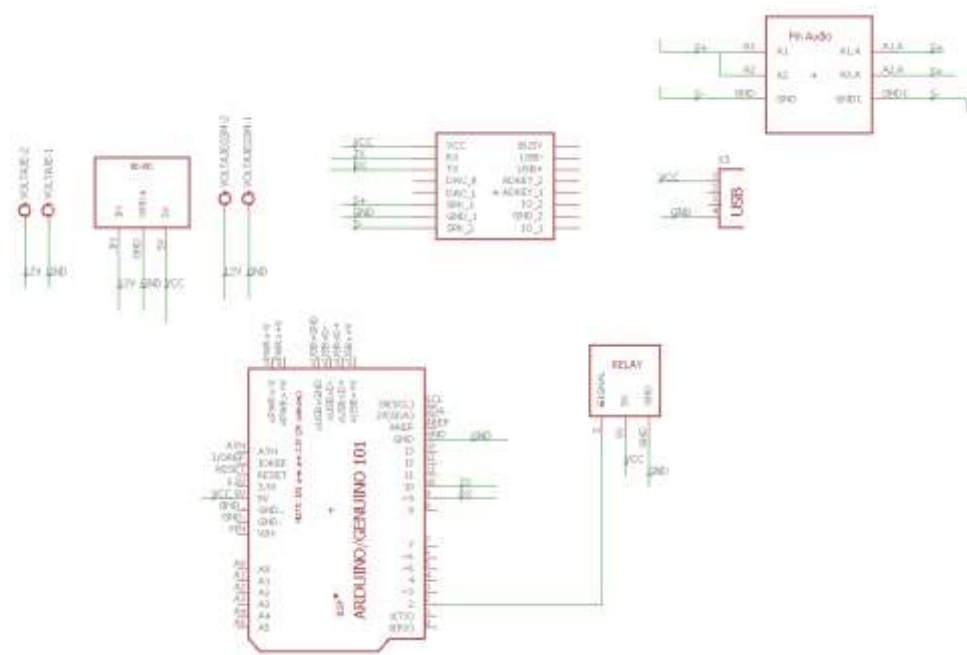


## Anexo G Prueba de detección de distancia en espacios no controlados





## Anexo H Esquemático de control



## Anexo I Esquemático de Board

